Lehrstuhl für Verteilte Informationssysteme
Fakultät für Informatik und Mathematik
Universität Passau

UNIVERSITÄT PASSAU

DOCTORAL THESIS

Annotated Interactive Non-linear Video

Software Suite, Download and Cache Management

Dipl.-Inf. Britta Meixner

October, 2014

Advisor: Prof. Dr. Harald Kosch
External Examiner: Prof. Dr. Maximilian Eibl

Abstract

Modern Web technology makes the dream of fully interactive and enriched video come true. Nowadays it is possible to organize videos in a non-linear way playing in a sequence unknown in advance. Furthermore, additional information can be added to the video, ranging from short descriptions to animated images and further videos. This affords an easy and efficient to use authoring tool which is capable of the management of the single media objects, as well as a clear arrangement of the links between the parts. Tools of this kind can be found rarely and do mostly not provide the full range of needed functions. While providing an interactive experience to the viewer in the Web player, parallel plot sequences and additional information lead to an increased download volume. This may cause pauses during playback while elements have to be downloaded which are displayed with the video. A good quality of experience for these videos with small waiting times and a playback without interruptions is desired.

This work presents the SIVA Suite to create the previously described annotated interactive non-linear videos. We propose a video model for interactivity, non-linearity, and annotations, which is implemented in an XML format, an authoring tool, and a player. Video is the main medium, whereby different scenes are linked to a scene graph. Time controlled additional content called annotations, like text, images, audio files, or videos, is added to the scenes. The user is able to navigate in the scene graph by selecting a button at a button panel. Furthermore, other navigational elements like a table of contents or a keyword search are provided. Besides the SIVA Suite, this thesis presents algorithms and strategies for download and cache management to provide a good quality of experience while watching the annotated interactive non-linear videos. Therefor, we implemented a standard-independent player framework. Integrated into a simulation environment, the framework allows to evaluate algorithms and strategies for the calculation of start-up times, and the selection of elements to pre-fetch into and delete from the cache. Their interaction during the playback of non-linear video contents can be analyzed. The algorithms and strategies can be used to minimize interruptions in the video flow after user interactions. Our extensive evaluation showed that our techniques result in faster start-up times and lesser interruptions in the video flow than those of other players. Knowledge of the structure of an interactive non-linear video can be used to minimize the start-up time at the beginning of a video while minimizing an increase in the overall download volume.

Kurzzusammenfassung

Moderne Web-Technologien lassen den Traum von voll interaktiven und bereicherten Videos wahr werden. Heutzutage ist es möglich, Videos in nicht-linearer Art und Weise zu organisieren, welche dann in einer vorher unbekannten Reihenfolge abgespielt werden können. Weiterhin können den Videos Zusatzinformationen in Form von kurzen Beschreibungen über animierte Bilder bis hin zu weiteren Videos hinzugefügt werden. Dies erfordert ein einfach und effizient zu bedienendes Autorenwerkzeug, das in der Lage ist, sowohl einzelne Medien-Objekte zu verwalten, als auch die Verbindungen zwischen den einzelnen Teilen klar darzustellen. Tools dieser Art sind selten und bieten meist nicht den vollen benötigten Funktionsumfang. Während dem Betrachter dieses interaktive Erlebnis im Web Player zur Verfügung gestellt wird, führen parallele Handlungsstränge und zusätzliche Inhalte zu einem erhöhten Download-Volumen. Dies kann zu Pausen während der Wiedergabe führen, in denen Elemente vom Server geladen werden müssen, welche mit dem Video angezeigt werden sollen. Ein gutes Benutzungserlebnis für solche Videos kann durch geringe Wartezeiten und eine unterbrechungsfreie Wiedergabe erreicht werden.

Diese Arbeit stellt die SIVA Suite vor, mit der die zuvor beschriebenen annotierten interaktiven nicht-linearen Videos erstellt werden können. Wir bilden Interaktivität, Nichtlinearität und Annotationen in einem Video-Model ab. Dieses wird in unserem XML-Format, Autorentool und Player umgesetzt. Als Leitmedium werden hierbei Videos verwendet, welche aufgeteilt in Szenen zu einer Graphstruktur zusammengefügt werden können. Zeitlich gesteuerte zusätzliche Inhalte, sogenannte Annotationen, wie Texte, Bilder, Audio-Dateien und Videos, werden den Szenen hinzugefügt. Der Betrachter kann im Szenengraph navigieren, indem er in einem bereitgestellten Button-Panel eine Nachfolgeszene auswählt. Andere Navigationselemente sind ein Inhaltsverzeichnis sowie eine Suchfunktion. Neben der SIVA Suite beschreibt diese Arbeit Algorithmen und Strategien für Download und Cache Management, um eine gute Nutzungserfahrung während der Betrachtung der annotierten interaktiven nichtlinearen Videos zu bieten. Ein Webstandard-unabhängiges Playerframework erlaubt es, das Zusammenspiel von Algorithmen und Strategien zu evaluieren, welche für die Berechnung der Start-Zeitpunkte für die Wiedergabe, sowie die Auswahl von vorauszuladenden sowie zu löschenden Elemente verwendet werden. Ziel ist es, Unterbrechungen zu minimieren, wenn der Ablauf des Videos durch Benutzerinteraktion beeinflusst wird. Unsere umfassende Evaluation zeigte, dass es möglich ist, kürzere Startup-Zeiten und weniger Unterbrechungen mit unseren Strategien zu erreichen, als bei der Verwendung der Strategien anderer Player. Die Kenntnis der Struktur des interaktiven nicht-linearen Videos kann dazu verwendet werden, die Startzeit am Anfang der Szenen zu minimieren, während das Download-Volumen nicht erhöht wird.

Contents

List of Figures xi

List of Tables xv

List of Listings xvii

List of Definitions xvii

1. Introduction 1
   1.1. Usage Scenarios with Interactivity and Non-linearity 4
   1.2. Problem Statement 7
      1.2.1. Creation and Playback of Annotated Interactive Non-linear Video . . 7
      1.2.2. User Experience during Playback 8
   1.3. Research Contributions 10
   1.4. Outline 11

2. Tools and Models for Interactive Non-linear Video 13
   2.1. Types of Extended Videos and Delimination 13
      2.1.1. Clickable Video 14
      2.1.2. Interactive Video 14
      2.1.3. Non-linear Video 16
      2.1.4. Multimedia Presentation 17
      2.1.5. Hypervideo 18
      2.1.6. Annotated Interactive Non-linear Video 18
      2.1.7. Overview and Summary 19
   2.2. Description Formats, Multimedia Models, and Standards 19
      2.2.1. Models and Languages for Interactive Multimedia Presentations . . 21
      2.2.2. Other Multimedia Presentation Models and Languages 24
      2.2.3. Models and Languages for Hypervideos 26
      2.2.4. Summary 28
   2.3. Evaluation of Existing Authoring Tools 29
      2.3.1. Other Authoring Tools 29
      2.3.2. Authoring Tools for Hypervideos 32
      2.3.3. Summary 36
   2.4. Evaluation of Existing Players 38
      2.4.1. Other Players 38
      2.4.2. Players for Hypervideos 41
      2.4.3. Summary 43

3. SIVA Suite 45
3.1. SIVA XML Schema: Modeling of Annotated Interactive Non-linear Video Behavior . . . 48
3.1.1. Conceptual Model . . . 48
3.1.2. XML Schema and XML File . . . 50
3.2. SIVA Producer: Management of Interactivity, Non-linearity, and Annotations in an Authoring Tool . . . 60
3.2.1. Non-linearity . . . 61
3.2.2. Annotations . . . 62
3.2.3. Interactivity . . . 64
3.3. SIVA Player: Realization of Interactivity, Non-linearity, and Annotations during Playback . . . 66
3.3.1. Non-linearity . . . 68
3.3.2. Annotations . . . 70
3.3.3. Interactivity . . . 71
3.4. Summary . . . 72

4. Techniques and Methods for Download and Cache Management 75
4.1. Player Implementations for Non-linearity in Videos . . . 75
4.2. User Behavior during Video Playback . . . 77
4.3. Scheduling of Download Queues . . . 80
4.4. Download and Streaming of Interactive (Non-linear) Video . . . 84
4.5. Cache Management and Replacement Strategies . . . 88
4.6. Summary . . . 90

5. Formalized Video Model, Hardware Constraints, and User Behavior 93
5.1. Video Model . . . 93
5.1.1. Annotated Interactive Non-linear Video . . . 94
5.1.2. Basic Functions and Definitions . . . 97
5.2. Hardware Constraints . . . 99
5.2.1. Cache Size . . . 99
5.2.2. Bandwidth . . . 100
5.3. User Behavior . . . 100
5.3.1. VCR Actions . . . 101
5.3.2. Extended Interactivity and Navigation . . . 101
5.4. Summary . . . 102

6. Download and Cache Management 105
6.1. Communication Architecture . . . 106
6.2. Global Calculations . . . 110
6.2.1. Sorting Elements of a Scene . . . 110
6.2.2. Calculation of the Starting Point . . . 110
6.3. Video Playback/Start Time Strategies . . . 114
6.4. Download Scheduling . . . 115
6.4.1. Constraints . . . 115
6.4.2. Pre-fetch Strategies . . . 117
6.4.3. Decision at Forks . . . 119
6.4.4. Download Strategy . . . 120

6.5. Delete Strategies . . . 120
6.5.1. Categories of Elements . . . 122
6.5.2. Indices for Deletion . . . 123
6.5.3. Delete Threshold . . . 125
6.6. Deadlocks . . . 125
6.7. Summary . . . 127

7. Evaluation 129
7.1. Performance Metrics . . . 130
7.2. Test Patterns and Test Graphs . . . 132
7.2.1. Patterns in Annotated Interactive Non-linear Videos . . . 133
7.2.2. User Generated Scenarios . . . 134
7.3. Test Configurations . . . 136
7.3.1. Description of the Videos and the User Behavior . . . 136
7.3.2. Parameters for the Algorithms/Strategies . . . 143
7.3.3. Description of the Environment . . . 143
7.4. Simulation Environment . . . 144
7.4.1. Internal Structure of the Framework . . . 145
7.4.2. Simulation Settings . . . 145
7.4.3. Implementation of the Framework . . . 146
7.5. Performance Evaluation . . . 148
7.5.1. Quality of the Test Results . . . 149
7.5.2. Evaluation of the Pre-fetch Strategies and Start Times . . . 156
7.5.3. Evaluation of the Delete Strategies . . . 169
7.5.4. Search for an “Optimal Combination” of Algorithms/Strategies for all Patterns . . . 175
7.5.5. Evaluation of the Strategies for Varying Numbers of Annotations . . . 181
7.5.6. Evaluation of the Strategies for Varying Pattern Widths . . . 187
7.5.7. Evaluation of the Strategies for the User Generated Scenarios . . . 190
7.5.8. Evaluation of the Strategies for Annotations with Varying Priorities . . . 194
7.6. Summary . . . 203

8. Conclusion and Future Work 205

A. List of Symbols 209

B. Standards and Models 213

C. Authoring Tools 223

D. Player 239

E. Sequence Chart of Interactions in the Framework 253

F. Settings of the User Generated Scenarios 255

G. Statistics for 1000 Test Runs on a Subset of Patterns and Settings 265

H. Multiple Comparison Test after Kruskal-Wallis Test for the 1000 Test Runs 279

I. Detailed/Further Statistics and Evaluation Results 287
I.1. Further Statistics 287
I.2. Evaluation of the Pre-fetch Strategies and Start Times 290
I.3. Evaluation of the Delete Strategies 302
I.4. Evaluation of the Number of Annotations 314
I.5. Evaluation of the Varying Path Probabilities and Pattern Widths 318
I.6. Evaluation of the Scenarios 322
I.7. Evaluation of the Priority-based Algorithms/Strategies 323

Bibliography 337

Index 387

List of Figures

1.1. Example of an annotated interactive non-linear video 2
1.2. Differences in the creation process of different types of video 3
1.3. Player with the virtual tour scenario 5
1.4. Player with the sport scenario 6
1.5. Connections between subproblems for download and cache management . 9

2.1. Different types of non-linear video 16
2.2. Overview over the different kinds of video 20
2.3. Chronology of description formats, standards, and models 28
2.4. Players categorized by their type and classification 44

3.1. Data exchange of the components of the SIVA Suite 46
3.2. Data flow in the SIVA Suite 46
3.3. Conceptual model of an annotated interactive non-linear video 49
3.4. Six parts of the SIVA XML format 50
3.5. Complex type “ProjectInformation” 53
3.6. Element “resources” 54
3.7. Complex types for static actions 55
3.8. Complex types for dynamic actions 56
3.9. Complex type “SceneList” 58
3.10. Complex type “TableOfContents” 59
3.11. Complex type “Index” 59
3.12. Screenshot of the SIVA Producer (overview) 63
3.13. Presentations of scenes in the scene graph 63
3.14. Screenshots of the SIVA Producer (annotation/table of contents editor) . . . 64
3.15. Annotation editor with hotspot editor 65
3.16. Selection panel in different players 68
3.17. Table of contents in different players 69
3.18. Search function in different players 69
3.19. Display of annotations in different players 70
3.20. Collaboration function in the HTML5 player 72

4.1. Different ways of implementing non-linearity in videos 76
4.2. Functions of different kinds of videos 77

5.1. Exemplary scene graph of a tour through a house 94
5.2. Exemplary schedule of one scene in detail 94
5.3. Example of an annotated interactive non-linear video 96

6.1. Component architecture and data flow of the video client 107
6.2. Simplified UML diagram illustrating the core of the simulation framework . 108
6.3. UML state diagram of the player framework 109

6.4. Playback and download schedule of a scene 111
6.5. Important timestamps in a download schedule 112
6.6. Illustration of the pre-fetch strategy PREFETCH_SL 118
6.7. Illustration of the pre-fetch strategy PREFETCH_FF 118
6.8. Assignment of delete categories for different allowed user behaviors 122
6.9. Example graph for the calculation of delete indices 125
6.10. A possible deadlock situation 127

7.1. Sequence pattern 133
7.2. Cycle pattern 134
7.3. Split/Join pattern 135
7.4. Mirrorworld/Counterpoint pattern 135
7.5. Sieve pattern 135
7.6. Probabilities for selecting a scene 139
7.7. Scenario C: scene graph and paths 141
7.8. Evaluation process 148
7.9. Statistics (distributions): frames to wait before playback 151
7.10. Statistics (distributions): number of pauses during playback 153
7.11. Statistics (distributions): download volume 155
7.12. Evaluation - pre-fetch (selected): frames to wait before playback/bandwidth 159
7.13. Evaluation - pre-fetch (selected): frames to wait before playback/cache size 159
7.14. Evaluation - pre-fetch (selected): waiting time before playback/bandwidth 161
7.15. Evaluation - pre-fetch (selected): waiting time before playback/cache size . 161
7.16. Evaluation - pre-fetch (selected): pauses during playback/bandwidth 163
7.17. Evaluation - pre-fetch (selected): pauses during playback/cache size 163
7.18. Evaluation - pre-fetch (selected): data vol. of elements not watched/BW . . 166
7.19. Evaluation - pre-fetch (selected): data vol. of elements not watched/CS . . 166
7.20. Evaluation - pre-fetch (selected): data volume of repeated downloads/BW . 168
7.21. Evaluation - pre-fetch (selected): data volume of repeated downloads/CS . 168
7.22. Evaluation - pre-fetch (selected): download volume/bandwidth 170
7.23. Evaluation - pre-fetch (selected): download volume/cache size 170
7.24. Evaluation - delete (selected): frames to wait before playback/bandwidth . 172
7.25. Evaluation - delete (selected): frames to wait before playback/cache size . 172
7.26. Evaluation - delete (selected): pauses during playback/bandwidth 174
7.27. Evaluation - delete (selected): pauses during playback/cache size 174
7.28. Evaluation - delete (selected): download volume/bandwidth 176
7.29. Evaluation - delete (selected): download volume/cache size 176
7.30. Evaluation - averages per pre-fetch/start-up 177
7.31. Evaluation - averages per delete strategy 179
7.32. Evaluation - weighted results per pre-fetch/start-up strategy 180
7.33. Evaluation - no. of annotations: frames to wait before playback/bandwidth 183
7.34. Evaluation - no. of annotations: waiting time before playback/cache size . 183
7.35. Evaluation - no. of annotations: pauses during scenes/cache size 185
7.36. Evaluation - no. of annotations: data volume of elements not watched/CS . 185
7.37. Evaluation - no. of annotations: download volume/cache size 186
7.38. Evaluation - pattern width: frames to wait before playback/bandwidth . . . 189
7.39. Evaluation - pattern width: data volume of elements not watched/bandwidth 189
7.40. Evaluation - pattern width: data volume of elements not watched/cache size 190

7.41.Evaluation - scenarios: frames to wait before playback 192
7.42.Evaluation - scenarios: waiting time before playback 192
7.43.Evaluation - scenarios: pauses during playback 193
7.44.Evaluation - scenarios: download volume 193
7.45.Evaluation - priorities (selected): frames to wait before playback/bandwidth 196
7.46.Evaluation - priorities (selected): frames to wait before playback/cache size 196
7.47.Evaluation - priorities (selected): waiting time before playback/bandwidth 197
7.48.Evaluation - priorities (selected): pauses during playback/bandwidth 197
7.49.Evaluation - priorities (selected): pauses during playback/bandwidth 199
7.50.Evaluation - priorities (selected): pauses during playback/cache size 199
7.51.Evaluation - priorities (selected): data vol. of elements not watched/BW . . 200
7.52.Evaluation - priorities (selected): data vol. of elements not watched/CS . . 200
7.53.Evaluation - priorities (selected): data volume of repeated downloads/BW . 201
7.54.Evaluation - priorities (selected): data volume of repeated downloads/CS . 201
7.55.Evaluation - priorities (selected): download volume/bandwidth 202
7.56.Evaluation - priorities (selected): download volume/cache size 202

E.1. Simplified UML sequence diagram of the simulation framework. 254

F.1. Task description for the graph creation task 256
F.2. Scenario A: scene graph and paths 257
F.3. Scenario B: scene graph 259
F.4. Scenario B: paths 260
F.5. Scenario D: scene graph and paths 262

I.1. Statistics (distributions): waiting time before playback 287
I.2. Statistics (distributions): data volume of elements not watched 288
I.3. Statistics (distributions): data volume of repeated downloads 289
I.4. Evaluation - pre-fetch (detailed): frames to wait before playback/bandwidth 290
I.5. Evaluation - pre-fetch (detailed): frames to wait before playback/cache size 291
I.6. Evaluation - pre-fetch (detailed): waiting time before playback/bandwidth 292
I.7. Evaluation - pre-fetch (detailed): waiting time before playback/cache size . 293
I.8. Evaluation - pre-fetch (detailed): pauses during playback/bandwidth 294
I.9. Evaluation - pre-fetch (detailed): pauses during playback/cache size 295
I.10. Evaluation - pre-fetch (detailed): data vol. of elements not watched/BW . . 296
I.11. Evaluation - pre-fetch (detailed): data vol. of elements not watched/CS . . 297
I.12. Evaluation - pre-fetch (detailed): data volume of repeated downloads/BW . 298
I.13. Evaluation - pre-fetch (detailed): data volume of repeated downloads/CS . 299
I.14. Evaluation - pre-fetch (detailed): download volume/bandwidth 300
I.15. Evaluation - pre-fetch (detailed): download volume/cache size 301
I.16. Evaluation - delete (detailed): frames to wait before playback/bandwidth . 302
I.17. Evaluation - delete (detailed): frames to wait before playback/cache size . 303
I.18. Evaluation - delete (detailed): waiting time before playback/bandwidth . . 304
I.19. Evaluation - delete (detailed): waiting time before playback/cache size . . . 305
I.20. Evaluation - delete (detailed): pauses during playback/bandwidth 306
I.21. Evaluation - delete (detailed): pauses during playback/cache size 307
I.22. Evaluation - delete (detailed): data volume of elements not watched/BW . 308
I.23. Evaluation - delete (detailed): data volume of elements not watched/CS . . 309
I.24. Evaluation - delete (detailed): data volume of repeated downloads/BW . . 310

I.25. Evaluation - delete (detailed): data volume of repeated downloads/CS . . . 311
I.26. Evaluation - delete (detailed): download volume/bandwidth 312
I.27. Evaluation - delete (detailed): download volume/cache size 313
I.28. Evaluation - no. of annotations: frames to wait before playback/cache size 314
I.29. Evaluation - no. of annotations: waiting time before playback/bandwidth . 314
I.30. Evaluation - no. of annotations: pauses during scenes/bandwidth 315
I.31. Evaluation - no. of annotations: data volume of elements not watched/BW 315
I.32. Evaluation - no. of annotations: data volume of repeated downloads/BW . 316
I.33. Evaluation - no. of annotations: data volume of repeated downloads/CS . . 316
I.34. Evaluation - no. of annotations: download volume/bandwidth 317
I.35. Evaluation - pattern width: frames to wait before playback/cache size . . . 318
I.36. Evaluation - pattern width: waiting time before playback/bandwidth 318
I.37. Evaluation - pattern width: waiting time before playback/cache size 319
I.38. Evaluation - pattern width: pauses during scenes/bandwidth 319
I.39. Evaluation - pattern width: pauses during scenes/cache size 320
I.40. Evaluation - pattern width: download volume/bandwidth 320
I.41. Evaluation - pattern width: download volume/cache size 321
I.42. Evaluation - scenarios: data volume of elements not watched 322
I.43. Evaluation - scenarios: data volume of repeated downloads 322
I.44. Evaluation - priorities (detailed): frames to wait before playback/bandwidth 323
I.45. Evaluation - priorities (detailed): frames to wait before playback/cache size 324
I.46. Evaluation - priorities (detailed): waiting time before playback/bandwidth 325
I.47. Evaluation - priorities (detailed): waiting time before playback/cache size . 326
I.48. Evaluation - priorities (detailed): pauses during playback/bandwidth 327
I.49. Evaluation - priorities (detailed): pauses during playback/cache size 328
I.50. Evaluation - priorities (detailed): number of skipped elements/bandwidth . 329
I.51. Evaluation - priorities (detailed): number of skipped elements/cache size . 330
I.52. Evaluation - priorities (detailed): data vol. of elements not watched/BW . . 331
I.53. Evaluation - priorities (detailed): data vol. of elements not watched/CS . . 332
I.54. Evaluation - priorities (detailed): data volume of repeated downloads/BW 333
I.55. Evaluation - priorities (detailed): data volume of repeated downloads/CS . 334
I.56. Evaluation - priorities (detailed): download volume/bandwidth 335
I.57. Evaluation - priorities (detailed): download volume/cache size 336

List of Tables

2.1. Overview description languages and authoring tools 37

3.1. Interactivity, non-linearity, and annotations in the SIVA Suite 47

4.1. Suitability of non-linearity forms for annotated interactive non-linear videos 76
4.2. Consideration of user behavior in related work 80
4.3. Characteristics of different scheduling algorithms in the domain of GPS . . 83
4.4. Related work on streaming and download of (non-linear) videos 87
4.5. Implementations of streaming proxies with cache replacement algorithms . 90
4.6. Basic and extended replacement policies for caches 91

5.1. Different intra-scene VCR actions 101
5.2. Interactive and navigational actions (intra-scene and inter-scene) 102

6.1. Calculated values for the proposed delete indices 126

7.1. Scenario C: settings 142
7.2. Relationships between the strategies for the WF_{start} metric 151
7.3. Relationships between the strategies for the WT_{start} metric 152
7.4. Relationships between the strategies for the P_{sum} metric 154
7.5. Relationships between the strategies for the $RDLV$ metric 154
7.6. Relationships between the strategies for the DLV metric 156

A.1. List of symbols 209

B.1. Standards/models for hypermedia applications 215
B.2. Standards/models for hypervideos 216
B.3. Standards/models for interactive multimedia presentations 217
B.4. Standards/models for multimedia presentations 222

C.1. Authoring tools for non-linear videos 224
C.2. Authoring tools for interactive videos 227
C.3. Authoring tools for hypervideos/hypermedia 230
C.4. Authoring tools for clickable videos 231
C.5. Authoring tools for multimedia presentations 237

D.1. Player for non-linear videos 240
D.2. Player for interactive videos 243
D.3. Player for hypervideos/hypermedia 246
D.4. Player for clickable videos 248
D.5. Player for multimedia presentations 252

F.1. Scenario A: settings 258

F.2. Scenario B: settings 261
F.3. Scenario D: settings 263

G.1. Statistics for 1000 test runs (WF_{start} metric) 267
G.2. Statistics for 1000 test runs (WT_{start} metric) 269
G.3. Statistics for 1000 test runs (P_{sum} metric) 271
G.4. Statistics for 1000 test runs ($DL_{not\ watched}$ metric) 273
G.5. Statistics for 1000 test runs ($RDLV$ metric) 275
G.6. Statistics for 1000 test runs (DLV metric) 277

H.1. Multiple comparison test after Kruskal-Wallis (WF_{start} metric) 280
H.2. Multiple comparison test after Kruskal-Wallis (WT_{start}) 282
H.3. Multiple comparison test after Kruskal-Wallis (P_{sum} metric) 283
H.4. Multiple comparison test after Kruskal-Wallis ($DL_{not\ watched}$ metric) 284
H.5. Multiple comparison test after Kruskal-Wallis ($RDLV$ metric) 284
H.6. Multiple comparison test after Kruskal-Wallis (DLV metric) 285

List of Algorithms

6.1. SortSceneLinear 111
6.2. GetStartFrame 114
6.3. SortVideoLinear 121

List of Listings

3.1. Example XML 51

7.1. Command line arguments of the getpoolsize mode 147
7.2. Command line arguments of the simulate mode 147
7.3. Command line arguments of the resume mode 147

List of Definitions

2.1. Definition (Clickable Video) 14
2.2. Definition (Interactive Video) 16
2.3. Definition (Non-linear Video) 17
2.4. Definition (Multimedia Presentation) 17
2.5. Definition (Hypervideo) 18
2.6. Definition (Annotated Interactive Non-linear Video) 19

5.1. Definition (Frame f) 95
5.2. Definition (Content of an Annotation α) 95
5.3. Definition (Priority of an Annotation Λ) 95
5.4. Definition (Annotation a) 95
5.5. Definition (Set of Frames $\mathcal{F}_{\mathcal{V}}$) 95
5.6. Definition (Set of Annotations $\mathcal{A}_{\mathcal{V}}$) 95
5.7. Definition (Scene p) 95
5.8. Definition (Start Scene p_{σ}) 95

5.9. Definition (End Scene p_ϵ) 95
5.10. Definition (Set of Scenes $\mathcal{P}_\mathcal{V}$) 95
5.11. Definition (Annotated Interactive Non-linear Video $\mathcal{V}$) 96
5.12. Definition (Set of Downloadable Elements $\mathcal{E}_\mathcal{V}$ of a Video) 98
5.13. Definition (Tuple of Frames of a Scene $\widehat{p}_x^f$) 98
5.14. Definition (Set of Frames of a Scene $\mathcal{F}_{p_x}$) 99
5.15. Definition (Set of Annotations of a Scene $\mathcal{A}_{p_x}$) 99
5.16. Definition (Set of Downloadable Elements $\mathcal{E}_{p_x}$ of a Scene) 99
5.17. Definition (Set of Successor Scenes $\mathcal{P}_{succ(p_x)}$) 99

6.1. Definition (Valid Schedule for one Scene $\mathcal{S}_{p_x}$) 116

1. Introduction

The history of moving pictures goes back to 1891 when the first film camera was invented by Thomas Edison [Mon09, p. 641]. The Lumière brothers projected movies onto a screen in 1895, which can be noticed as the birth hour of cinemas [Mon09, p. 641]. Many improvements in transmission and viewing technology were made until the NTSC system, the first of three systems of color television was invented in the USA in 1953 [Gul07, p. 2]. The reception of films was purely passive for the viewer in that time. It was not possible to influence the film in any way after selecting a TV channel or a cinema hall. The invention of VHS for home video recording in 1976 [Shi85] can be seen as the first type of time shifted TV. The viewer was able to fast-forward and fast rewind which allowed to start viewing anywhere in between the video. This was an increase in the interactivity compared to the reception of a previously selected program on TV. Still, the different types of media (text, film, images) were strictly separated from each other in their usage and presentation. The usage of (internet) connected TV grew with increasing bandwidths in the Internet. Thereby, different technologies arose, like HbbTV and IPTV. Trends go to digital viewing on demand and "due to mobile devices, [it was] not only about time-shifting but place-shifting as well" [IBM11]. HbbTV and Web-TV/IPTV provided higher levels of interactivity like program selection, simple interactive questions, and background explanations [Loh09]. But in Web-TV/IPTV no constant QoS could be provided as in traditional TV. Jain and Wakimoto stated already in 1995 that "with the increased bandwidth, and advances in several areas of technology, the time [had] come to address issues involved in providing real interactive video and TV systems" [JW95].

The usage of TV and VHS allowed only limited interactivity because of the used hardware with a missing reverse channel. An important milestone towards interactive videos and multimedia was made in August 1981 when the microcomputer IBM-PC with the operating system MS DOS was developed [Mon09, p. 658]. According to Smith, "Three major sectors use[d] and stud[ied] interactive video: the military and government, private industry, and education" in the early 1980s [Smi87]. The computer readable CD with an appropriate format for videos and a storage capacity of 650 MB was released in 1984/85 [Pee10]. "In 1986, it was announced that CD-I (compact disk-interactive) specifications would be submitted, so that CD-I players could be developed for the consumer market that would handle compact disks with video, images, graphics, audio files, text, data – as well as software to support interactive use" [Fox89]. These innovations made the first multimedia CD called "Companion to Beethoven's Ninth" in November 1989 possible. The first multimedia CD containing a whole film, "A Hard Day's Night", followed in 1993 [Mon09, p. 663].

The first hyperlinked videos "Video Finger" [Wat87] and "Elastic Charles" [BD89] were created by the MIT Media Laboratory. One of the earliest research papers on hypervideo was published by Sawhney, Balcom, and Smith in 1996. After creating a hypervideo prototype called HyperCafe with Macromedia Director 4.04 they proposed the implementation of a tool for "hypervideo authoring and navigation" [SBS96]. Furthermore, suggestions were given for the functions of such a tool. A comparably high level of interactivity was reached combining

navigational elements with different types of media. An early definition of interactivity in videos was presented by Stenzler and Eckert. They said, "A video application is interactive if the user affects the flow of the video and that influence, in turn, affects the user's future choices" [SE96]. In the early 90s portable storage capacities were as limited as the transfer rates on the Internet. In this context, the first software for video editing "QuickTime" was published by Apple in 1991 [Mon09, p. 662]. Only after the invention of DVDs in 1997, the storage of up to 4.7 GB of data was possible and allowed the user to store videos from 1.0 to 9.0 hours [Tay99]. As new form of interactivity, navigation menus became available on each DVD allowing to jump into scenes instead of fast-forwarding. The presentation of interactive videos as an overall concept in the Internet was not possible at that time, because continuous media (like audio files and videos) could not be played directly (inline) on a web page [WFW96].

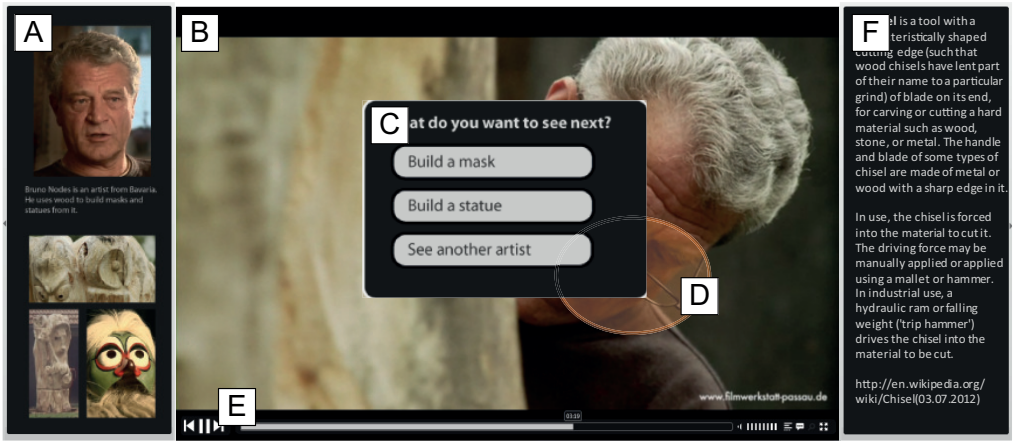


Figure 1.1.: Example of an annotated interactive non-linear video: (A) annotation area with images and text, (B) main video, (C) selection panel, (D) clickable area, (E) extended time line and control panel, (F) annotation area with text.

According to Shipman, Girgensohn, and Wilcox "[...] the growing use of digital cameras (video and "still") to capture short video snippets makes home authoring of interactive video an application that is likely to emerge" [SGW03c]. Furthermore, the number of mobile devices has grown in the last years and will grow further according to the Cisco Visual Networking Index (VNI) [Cis14]. "Because mobile video content has much higher bit rates than other mobile content types, mobile video will generate much of the mobile traffic growth through 2018. [...] Of the 15.9 exabytes per month crossing the mobile network by 2018, 11 exabytes will be due to video. Mobile video represented more than half of global mobile data traffic beginning in 2012, indicating that it is having an immediate impact on traffic today, not just in the future" [Cis14]. With new technologies and improvements in standards, transmission hardware, processors, internal storage, new methods of programming, and reliable software libraries, it is possible to provide high levels of interactivity in multimedia contents nowadays. A special form of multimedia content is interactive (non-linear) videos which are based on videos and provide a wider range of interactivity than traditional linear videos. They can be described as follows: annotated interactive non-linear videos always consist of a main video scene (Figure 1.1, (B)) combined with other elements (annotations). The main video scenes are linked to each other in a non-linear way. Forms of annotation that can be added to these scenes are plain text, rich text, hyperlinks, images, audio, or video files (Figure 1.1, (A), (F)). The different kinds of annotation can be divided into two subgroups. One group is continuous

annotations like audio files, videos, or animations. The other group is static annotations like text or images. At different points in time in the main video, annotations are displayed or hidden. Continuous annotations may require synchronization with the main video. Furthermore, different forms of navigation can be implemented in annotated interactive non-linear videos. A scene graph connects scenes of a video, describes the course of the video, and can be used to implement a non-linear flow. The viewer may receive the scene graph as a navigational element in the player or just get buttons (Figure 1.1, (C)) to chose what he wants to see next at the end of a scene. Quizzes can be used to test the knowledge of a viewer and link to different scenes according to a quiz result. A table of contents can be created that links directly to scenes. The viewer is able to jump to a scene in the video without watching the whole video or searching for the point in time by fast-forwarding. A keyword search can be useful to search for scenes with a particular content. Keywords may be added by the author of an annotated interactive non-linear video manually or in an automated way. Clickable objects (Figure 1.1, (D)) in the video can show information about the clicked object.

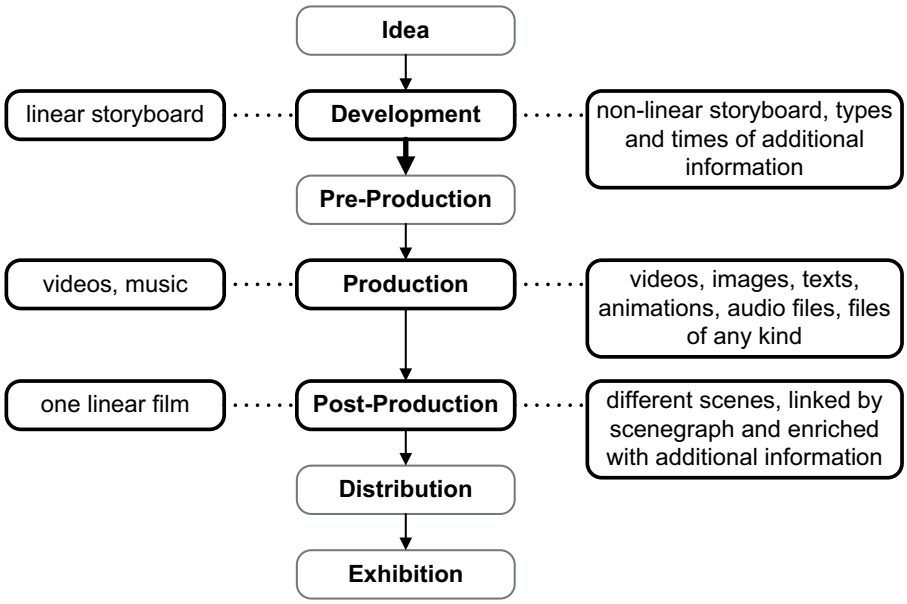


Figure 1.2.: Differences in creating a traditional linear video and in creating an annotated interactive non-linear video (inspired by [Wor09]).

The planing and production workflow for annotated interactive non-linear video is more complex than for traditional linear video (see Figure 1.2). Tools and software for pre- and post-production as well as the production equipment are needed to create high quality video contents. Furthermore, software for the creation of the non-video annotations is needed. All produced media materials have to be linked to an overall presentation as a last step in the post-production. Therefore, an authoring software is needed which provides easy-to-use functions to create a scene graph for the non-linear flow, to define interactivity at certain points in time, and to link the annotations to the scenes. The annotated interactive non-linear videos are displayed on desktop PCs and mobile devices with special players. These players have extended time lines/control panels showing annotation markers and buttons to open the search-function or the table of contents. Collaboration functions can be one step going further than the described interaction with the video. The user can access these functions in

an extended control bar in the player, offering more buttons than the commonly known player controls.

1.1. Usage Scenarios with Interactivity and Non-linearity

The removal of technical barriers opens a range of use cases for annotated interactive non-linear videos from simple clickable videos for marketing purposes to highly enriched, linked, and interactive videos. These videos are used in a variety of scenarios. Complex scenarios with higher levels of interaction can be found in virtual tours, mobile help systems, e-learning, sport events, interactive video stories, and memory training which are described in detail hereafter.

Virtual tours range from walks through a house (Figure 1.3) or single building like a museum [Got06; MPG07] to tours through whole cities. A tour through a house can be used by architects or manufacturers to show prefabricated buildings to potential customers. The information provided thereby may range from descriptions of pieces of furniture to the description of building materials. Furthermore, detailed information about the sizes of single rooms or prices are provided with the video. Further impressions of a building can be given by images. These may show perspectives which are not covered by the main video or detailed views of objects in the building. A tour through a museum may provide multimedia enriched background information of the exhibitions. If the virtual tour is implemented for a group of buildings, it might be used as a guide. Opening hours of offices and information about different rooms or people working in the offices can be provided. The tour might also be implemented for a part of a city, for example a historic city center. Then, detailed information like the history of a building, pictures of festivals, or details of buildings as well as opening hours, contact information, and ratings of hotels, restaurants, and shops can be provided. These examples could be extended to tours through whole cities. Thereby, more general descriptions of districts, traffic junctions, or information about the whole town can be added to the video.

Interactivity can also be integrated in different types of **sport videos**. Training videos can be annotated with information about errors. With a collaborative editing function, trainers in different locations can discuss issues and possible improvements of an athlete's motion sequence (as described in [Sin+11]). Videos of sport events are usually filmed from different perspectives. Users can chose their own perspective for viewing the event. Furthermore, it is possible to search for an athlete or a certain scene, if the video is indexed appropriately. Larger events that last several days can be divided into days or single competitions which can be accessed by using a table of contents. An example of one run in a mounted archery competition can be found in Figure 1.4.

Interactive video stories, like crime movies, may have several story lines. The viewers could decide which turn the plot will take. Furthermore, it would be possible to include the decisions of the viewer into the selection of the following scene. Implementing that, the viewer is able to solve the mystery in a kind of interactive quiz. Therefor, the linear flow of the video has to be broken up by providing navigational elements like a selection panel with buttons or a quiz. Besides the non-linear composition of scenes, a parallel display of the main video and additional information (annotations) can enrich and extend the viewing experience in a different way. Detailed descriptions of objects may help the viewer to make the correct decisions while trying to find the perpetrator. Intra-scene navigation can be extended by clickable markers when an annotation is displayed, so the viewer can find information more

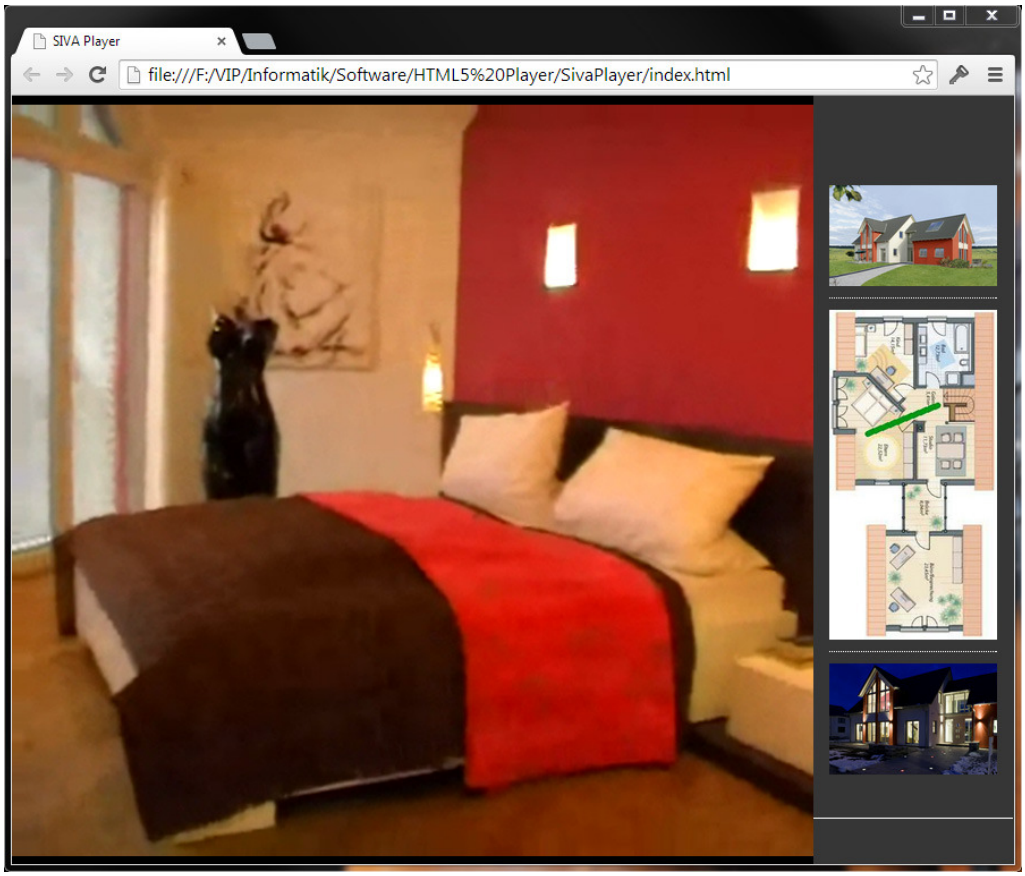


Figure 1.3.: Player with the virtual tour scenario: walk through a house, main video with annotation area.

quickly. A second video can be displayed with the main video to show a second view on an object in the video or the perspective of a second actor. If objects in the video are clickable, the user might explore more detailed information on the object or on circumstances related to the object helping him with his inquiry. In more complex videos it might be useful for the viewer to jump back to a certain scene or annotation. Therefor a keyword search and/or a table of contents can be provided showing only those scenes that have been discovered before. This use case was described by Meixner and Kosch in [MK13], where a more detailed description and analysis can be found.

While videos and other types of media are widespread in **e-learning**, annotated interactive non-linear videos can be found rarely. Zhang et al. carried out a study that supported their hypothesis "on the positive effects of interactive video on both learning outcome and learner satisfaction in e-learning" [Zha+06]. A study on using hypervideo presentations for teaching and learning was carried out by Mujacic et al. in [Muj+12] which confirms this result. Lusk et al. claim that "the segmentation of multimedia instruction facilitates basic (recall) and deep (application) knowledge acquisition" [Lus+09]. They furthermore show that there is an "individual difference variable that affects learning in a multimedia instructional environment" [Lus+09]. Hasler, Kersten and Sweller state, that "Learner control, either in the form of predefined segments or by allowing the learners to pause the animation at any time, should be integrated in educational animation in order to improve instructional efficiency" [HKS07]. This result is confirmed by Spanjers et al. who conclude "that both segmentation by cueing in

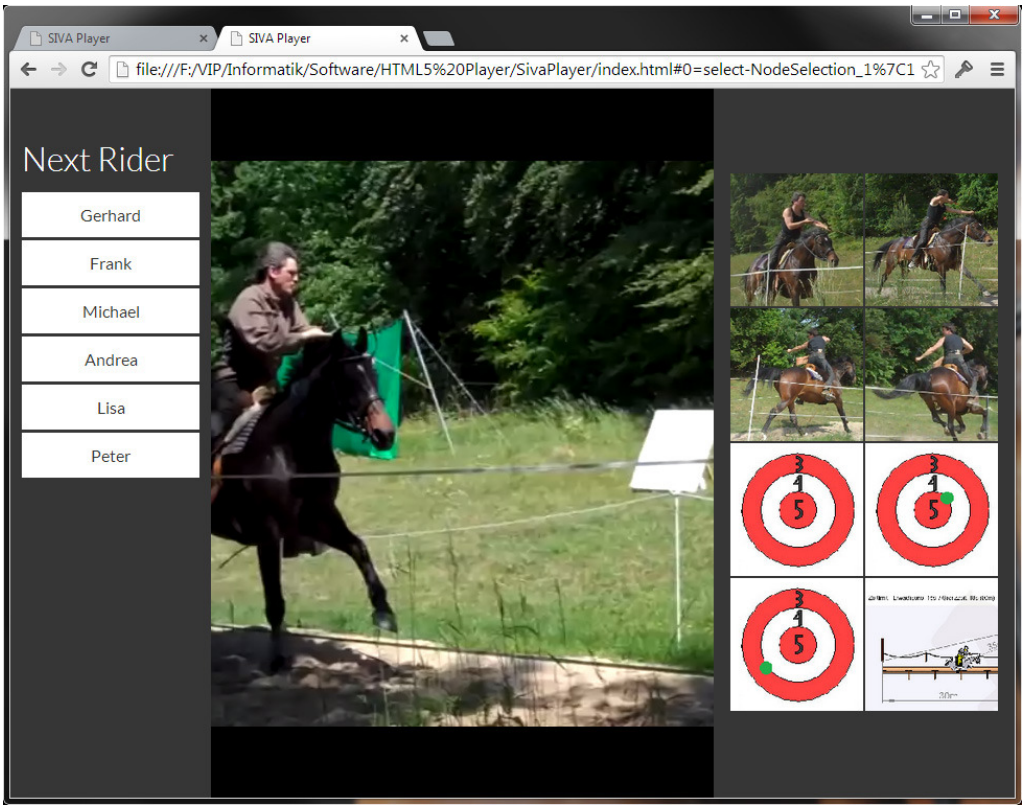


Figure 1.4.: Player with the sport scenario: clips of single runs of a competition, main video with annotation area on the right and selection panel on the left.

the form of temporarily darkening the screen and by pausing have a positive effect on learning outcomes or cognitive load" [Spa+12]. A comprehensive survey on multimedia authoring tools for educational purposes was conducted by Kaskalis, Tzidamis, and Margaritis in 2007 which revealed that no suitable tool could be found back then [KTM07]. Nevertheless, interactive videos are suitable to address all requirements emerging from this area. The video and additional information provided with it can be adjusted to the knowledge of the learner. This knowledge is determined by tests. According to the answers in a test, a repetition of a whole section is provided or future materials are enriched with additional information. Learning paths are defined by teachers and activated corresponding to the current knowledge of the learner during playback. A table of contents provides an overview of the structure of the lessons and can be used to navigate to a certain section for repetition.

Mobile help systems can be realized with annotated interactive non-linear videos using comparably cheap mobile end user devices like tablets or smart phones available these days. Especially in manufacturing companies, mobile help systems are more effective because of the time-independence of the knowledge transfer. It is possible to add additional information like images or other videos of different perspectives of a machine to an instructional video. The implementation of guided troubleshooting is accomplished by building up a multi-stage decision-making process which leads to a detailed video-instruction about how to solve a problem. The level of detail can be adapted to meet the viewers needs. For example, different additional information is provided if the video is watched by a worker or a repairer. An extension of predefined annotated interactive non-linear video with new collaborative elements might enable a company to build up a wider and more detailed pool of information which

makes the instructions easier to understand. A more detailed description and analysis of this use case is described by Tonndorf et al. in [Ton+12].

Furthermore, **memory training** is a use case which can be realized with annotated interactive non-linear videos very well. They can be implemented as a labyrinth or a quiz, where the viewer has to evoke certain contents, actions of actors, or objects in a scene. Depending on the selection or quiz result, the follow up scene is selected. Thereby, an increase in difficulty or a variation of tasks may be implemented with the underlying non-linear structure. Interactive elements can be used to make a labyrinth more interesting for the viewer. Furthermore they can be used to encourage the viewer to interact with the video.

1.2. Problem Statement

While a self-explanatory GUI makes annotated interactive non-linear videos producible for non-professionals in the authoring tool, download and cache management at player side is far more critical. Usually, only few authors are using the authoring tool, but the outcome, the annotated interactive non-linear video, is watched by lots of (Internet) users expecting a high quality of experience[1]. The two major problems this work is dealing with are the implementation of a software suite and the development of a download and cache management.

1.2.1. Creation and Playback of Annotated Interactive Non-linear Video

A software suite containing an authoring tool, a data exchange format, and one or more players is needed. Keeping track of all elements that an annotated interactive non-linear video contains can be challenging in larger video projects. The authoring tool has to provide as much help as possible during the creation process. Furthermore, the authoring tool has to be able to deal with different video, audio, and image formats. These have to be converted into the formats specified for the various players. A way to transfer control information and data from an authoring tool to a player is to save it in an XML file. Media files which are referenced by the control file are stored in a folder structure. The video is then downloadable to a device and can be played without Internet connection. The file must have a well defined file structure in order to make the processing effective at player side. Different standards like SMIL [W3C12], NCL [Tel11], and HTML5 [W3C13c] exist. Either these standards have to be extended because none of them offers the whole range of functions needed for annotated interactive non-linear video, or a new XML structure has to be created, which should fulfill the following requirements: The format is based on a scene graph and annotations are triggered by scenes or users. Local and global, as well as different kinds of annotations exist. A table of contents and keyword search represent secondary navigation structures. The flow-control is event-based and timing issues should be kept as local as possible. Status information can be stored and evaluated. While more and more different end-user devices have Internet access, the technical characteristics like display sizes, input devices, and the speed of the Internet access of these devices vary widely. In addition, player implementations for different platforms

[1] "QoE can be considered [as] the semantic variant of QoS since, broadly speaking, it denotes the overall experience that is witnessed by an end-user. Stated differently, it refers to a consumer's satisfaction when using a particular product or service." [Wij+11]

may be necessary. Suitable control elements have to be provided for the extended functions of annotated interactive non-linear video players.

Dealing with the described problems, this dissertation attempts to answer the following research questions in the area of creation and playback of annotated interactive non-linear videos:

- What different types of extended videos do exist and how can the terms be deliminated from one another?
- Is one of the existing description formats capable of describing interactivity, non-linearity, and additional information for the proposed type of video?
- Which authoring tools and players do already exist and what are their shortcomings with regard to usability for annotated interactive non-linear videos?
- How can content and control information be modeled for playback?
- How can the composition of interactivity, non-linearity, and additional information be comprehensibly managed in an authoring tool?
- How are interactivity, non-linearity, and the display of additional information realized in desktop and mobile players?

1.2.2. User Experience during Playback

Dealing with a traditional linear video, the estimated download or buffer duration can be calculated at a given bandwidth. After that, the point from which the video can be played without interruption can be concluded. The user is allowed to interact with the player controls and is able to jump forwards and backwards in a video or pause the video. This makes estimations on buffer durations much more complex. The estimations on buffer durations is even more complex for annotated interactive non-linear videos, where an extended set of interactions is allowed on a non-linear structure of scenes. The structure of the annotated interactive non-linear video leads to different problems during playback, if the video cannot be downloaded to the playback device as a whole (for example because of insufficient storage capacities or high resolution images and videos). Breaks may occur as a result of loading times for new contents or for loading a new web page as in the case of YouTube Video Annotations [You13]. Interruptions in the video flow, after user interactions, destroy the perception of a single video and decrease the user's quality of experience. Concerning one single video, the findings of Hossfeld et al. [Hos+12], Egger et al. [Egg+12], and Krishnan and Sitaraman [KS12] show that stalling during video playback has a very bad influence on the quality of experience while watching a video. Most users prefer initial delays which should not be too long to avoid abandonment. The initial delays and stallings have to be minimized by using suitable download, caching, and delete strategies. While developing these strategies, several internal and external conditions have to be taken into account. Additional information like images, audio files, and videos enhance the download volume of a scene. This additional amount of data has to be factored into the time calculations for download and playback. Parallel story lines increase the download volume for future scenes, because parts of all parallel story lines have to be downloaded to minimize waiting times after selecting a scene. A logic implemented in fork nodes may enable or disable paths of the video, which limits the options for the algorithms on the one hand, but requires an evaluation logic on the other hand. In

addition, different users may behave differently in varying video-structures like virtual tours or e-learning. If behavior patterns can be identified, it could be possible to optimize the download, caching, and delete algorithms and strategies. Different end user devices (smart phone, laptop, PC) provide different cache and bandwidth capabilities. Therefore algorithms and strategies have to be robust with respect to small cache sizes and/or low bandwidths. It might be necessary to develop special algorithms/strategies for these settings which allow a more flexible handling of the cache. Furthermore, only parts of the video are needed at the end-user device, because not all scenes and annotations may be watched by the end user.

Our approach is to split the problem of providing a good user experience by implementing a download and cache management into smaller parts which than can be solved individually and reassembled afterwards. One problem is that non-linearity has to be implemented in a traditionally linear player paradigm. Furthermore, new and more forms of user interaction are possible in annotated interactive non-linear videos compared to traditional videos. A path has to be selected at a fork in the video to continue with the video. Different decision criteria may be applied to make a choice on what follow-up scene should be selected. While these forms of interaction need to be implemented in the player GUI, contents need to be downloaded from the server and cached on the client. Each scene has to be scheduled for download in order to provide all elements at client-side at the right time. After a schedule is created, protocols and strategies need to be applied to transfer all data from server to client. Already downloaded data should be saved in the client cache as long as possible. When the cache is full, strategies have to be applied which delete those elements that are not needed any more. Figure 1.5 shows the connections between the problems described so far. As illustrated, each separate area of related work is connected with at least two others. Used algorithms and strategies need to fit together for that reason.

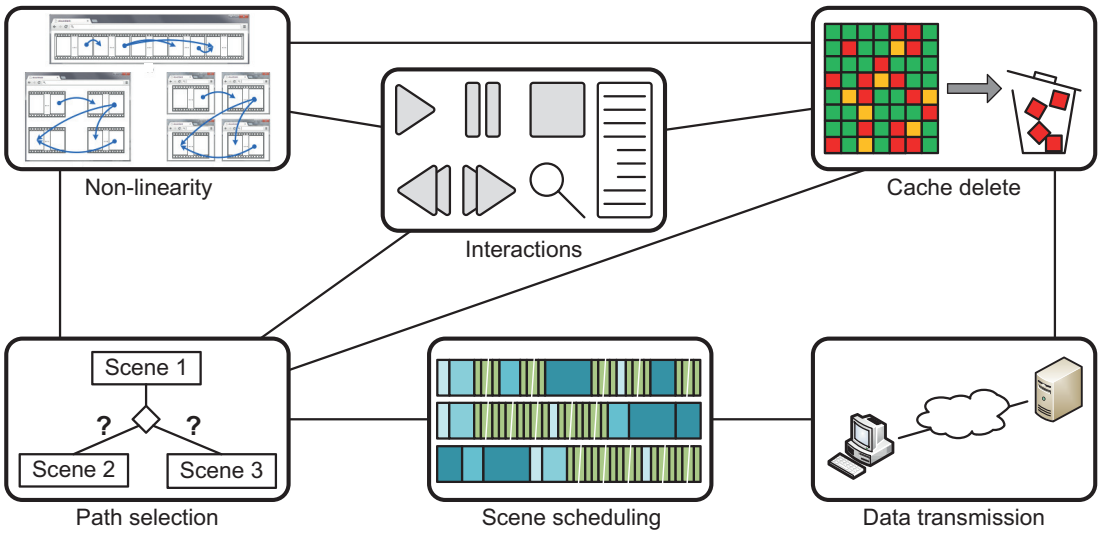


Figure 1.5.: Connections between single subproblems in download and cache management of annotated interactive non-linear videos.

Varying programming languages in player implementations allow different levels of influence on download and cache management of continuous media. In order to be able to provide comparable results between developed algorithms and strategies and to avoid multiple implementations in different programming languages, a modular simulation framework is desired. This allows a restriction free implementation of the designed algorithms and strategies.

Trying to find a solution for the described subproblems which fits together to an overall solution, this thesis attempts to answers the following research questions:

- What does the communication architecture of annotated interactive non-linear videos look like and how do single components interact in it?
- How can a starting point for playback, which avoids interruptions, be calculated?
- How can a start-up delays be reduced?
- How can the elements be scheduled for download from the server to the client?
- By which criteria are elements deleted from the cache?
- Are the results of the simulations significant enough to derive statements from only one run of each test in a simulation?
- Which combination of algorithms/strategies results in the smallest number of frames to wait before playback averaged over all patterns?
- Which combination of algorithms/strategies results in the smallest waiting times before playback averaged over all patterns?
- Which combination of algorithms/strategies results in the fewest pauses during playback averaged over all patterns?
- Which combination of algorithms/strategies results in the smallest data volume of not watched elements averaged over all patterns?
- Which combination of algorithms/strategies results in the smallest data volume of repeatedly downloaded elements averaged over all patterns?
- Which combination of algorithms/strategies results in the smallest download volume averaged over all patterns?
- How do selected combinations of algorithms/strategies perform in more extreme settings (more annotations, wider patterns, disadvantageous path probabilities)?
- Do selected combinations of algorithms/strategies also show corresponding results in real world scenarios compared to the results from the patterns?

Parts of this subsection (1.2 Problem Statement) were taken and adapted from our previous works [Mei+12b], [MH12], and [MK12].

1.3. Research Contributions

Trying to provide working and evaluated solutions for the tasks and questions described in Section 1.2, this work makes the following two major research contributions:

Software Suite and Underlying Structures: An event-based modular **XML file specification** is proposed to bring the **formal model** of the term "annotated interactive non-linear video" into a transferable form authoring tool to player. A conceptual model is used to clarify the definition. Both, XML file and formal model outline the relationships between the various media elements. Furthermore, rules are defined to specify inter-

actions and the display of scenes and annotations. Paths of an annotated interactive non-linear video are disabled or enabled by a **user interaction**. An **authoring tool** called SIVA Producer is presented. It is capable of accomplishing all steps from video editing and creating the video structure, the annotation of scenes, and a table of contents up to annotations for content search in the video. Results of usability and end user tests are presented to show the simplicity of use. A **web-player** and **two mobile players** are introduced. The web-player has functions for logging as well as a collaboration feature, which enables a user to add text and images to the video. Mobile players are implemented as apps for the Android platform and for iPads, iPods, and iPhones. Solutions are designed on how the display can be split or how the viewer can be alerted if an annotation is available (if not displayed). A library function allows the user to download the videos to watch them later when no connection to the Internet is available.

Download and Cache Management: A **modular player simulation framework** was implemented to be able to find appropriate solutions for different end user devices and viewer behavior. The essential part is the player logic. It allows a download and cache management that is able to adapt to viewer behavior and underlying hardware. The **download scheduling** consists of pre-fetching strategies, a decision logic for forks, and a download strategy. The latter combines created download queues, relative frequencies/probabilities, and constraints. It downloads all parts of the annotated interactive non-linear video required for playback, according to the structure of the video and the estimated user behavior. Pre-fetching strategies, decision logic for forks, and download strategies have a modular design to make them exchangeable when different strategies are designed. Furthermore, we propose a **cache management** strategy. The point in time for clearing the cache is definable as well as the amount of data that should be deleted. Delete strategies are implemented which weight the elements in the cache based on different criteria. These algorithms and strategies allow the caching of videos and annotations until they are needed. The contents of the cache are managed when erasing procedures are executed if space is needed. The synchronization of video and annotations during playback is provided to ensure realistic simulations.

Parts of this subsection (1.3 Research Contributions) were taken and adapted from our previous works [MKK11], [Mei+12b], [MH12], [MK12], and [Mei+13].

1.4. Outline

This document contains an introduction, two major parts consisting of six chapters, and a conclusion. The first major part describes the software (SIVA Suite). It gives an overview of related work. Related authoring tools, multimedia models, and players are evaluated. Furthermore, the SIVA Suite, a framework for annotated interactive non-linear videos consisting of an authoring tool, an XML file for control data, and different players for mobile and desktop environments is described. The second and main part of this work describes the download and cache management for annotated interactive non-linear video players. Its efficiency is evaluated using a specially developed framework. Different evaluations are performed with the framework and the results are presented. The following chapters are described in detail as follows:

Chapter 2 presents the differences between the terms “clickable video”, “interactive video”, “non-linear video”, “hypervideo”, “multimedia presentation”, and “annotated interactive non-linear video”. It narrows down the kind of video this work is dealing with. Tools and models for annotated interactive non-linear videos are described and compared. Thereby, their strengths and weaknesses are pointed out.

Chapter 3 introduces the SIVA Suite, a framework for annotated interactive non-linear videos consisting of an authoring tool, an XML structure and conforming XSD, and different players. A metadata model for annotated interactive non-linear videos is presented whereof an XSD is derived. The implementation of interactivity, non-linearity, and additional information in the authoring tool – the SIVA Producer – and its components are described in detail. Furthermore, the realization of interactivity, non-linearity, and additional information in an HTML5 web player with collaborative elements and a logging functionality is described as well as two mobile players, one for the Android and the iOS platform.

Chapter 4 compares existing techniques and methods for download and cache management. The behavior of players for this type of media is examined. Approaches taking the user behavior into account for download or streaming of linear videos are analyzed for their suitability for the proposed approach. Furthermore, solutions for download and streaming of interactive (non-linear) videos are studied and evaluated. Existing cache management and replacement strategies from different areas are evaluated for their usage in annotated interactive non-linear videos.

Chapter 5 describes influences which affect annotated interactive non-linear videos. A differentiation between the video model, the hardware constraints, and the user behavior is made. Each category is subdivided, specified, and described.

Chapter 6 deals with download and cache management for annotated interactive non-linear videos. First, a communication architecture is introduced. Then, global calculations are presented and different playback scenarios are described. The download scheduling part is divided into the four subparts which have to be taken into account. Constraints have to be fulfilled while pre-fetch strategies are designed. A download strategy schedules the elements of previously arranged queues. Delete strategies decide which elements have to be erased from the cache, if space is needed for new elements. Furthermore, a strategy to avoid deadlocks is introduced.

Chapter 7 evaluates the download and cache management strategies from Chapter 6 for their applicability for integration into the players. Therefor, performance metrics are defined and justified for their use in the evaluation. A modular simulation environment for the evaluation of the previously described algorithms and strategies and their combinations is introduced and explained. Pattern-based test configurations are stated precisely. Performance evaluations testing pre-fetch strategies, start times, and delete strategies are analyzed and compared for their suitable areas of use. Furthermore, selected strategies are tested with varying numbers of annotations, pattern widths, path probabilities, and annotation priorities. User generated scenarios are taken into consideration for selected strategies as well.

Chapter 8 summarizes the findings and contributions of this work. The influences of download and cache management strategies on other video players and existing standards are analyzed. An outlook and possible future developments conclude this work.

2. Tools and Models for Interactive Non-linear Video

This section gives an overview of related work from the areas “clickable videos”, “interactive videos”, “non-linear videos”, “hypervideos”, and “multimedia presentations”. Thereby, it tries to answer the following research questions:

> *What different types of extended videos do exist and how can the terms be deliminated from one another?*
>
> *Is one of the existing description formats capable of describing interactivity, non-linearity, and additional information in extended videos?*
>
> *Which authoring tools and players do already exist and what are their shortcomings with regard to usability for interactive non-linear videos with additional information?*

The terms “clickable video”, “interactive video”, “non-linear video”, “hypervideo”, “multimedia presentation”, and “interactive non-linear video with additional information” are differentiated with a literature review and a summarizing definition for each term. Subsequently authoring tools are categorized by these definitions and analyzed for their usefulness as tools to create interactive non-linear videos with additional information. Models and languages are also evaluated for their use. Existing players and parts thereof from different areas are checked for their applicability to interactive non-linear videos with additional information.

2.1. Types of Extended Videos and Delimination

Different terms like “clickable video”, “interactive video”, “non-linear video”, “hypervideo”, “multimedia presentation”, or “annotated interactive non-linear video” appear in the literature. Some of them are not used in a consistent way throughout the related work. This section compares the usages of the different terms and states a working definition for each of them. Thereby, the term *element* is used to describe an arbitrary object or a person in the video. An *annotation* is an additional information displayed with a main medium. The medium of an annotation may be a text (plain or formatted), a picture, a graphic, a video, an audio file, an animation, or any other kind of medium that can be shown in a player. A summarizing graphic of the definitions given hereafter is presented in Section 2.1.7. A categorization with different criteria can be found in [Mat11].

Related work and tools from the areas of *video browsing* like vizard[1], *video search* like YOVISTO[2] [SW10; WLS11], *multi-view video* [Kel+95; Kat+96; MF11; Mil+11; XCL12], tools

[1] `http://www.video-wizard.com/index-n.htm` (accessed April 26, 2014)
[2] `http://www.yovisto.com/` (accessed April 26, 2014)

for *video annotation using video analysis* like longomatch[3], m-ontomat-annotizer [Pet+06], or iVAT [Bia+13], and *social video/social TV* [Wan+12; Shi13] are not taken into account in this work. All of them are somehow related to interactive non-linear videos with additional information. They furthermore provide certain interactive or non-linear features, but they are not as closely related as the types of extended videos described in this section.

2.1.1. Clickable Video

Clickable videos are the simplest type of enriched videos. No scientific definition of the term could be found to the best of our knowledge, but the term is commonly used in currently available players or websites: Internet users are allowed to click on elements in the video. Clickable elements are called hotspots [Cli12], the whole technology of clicking on an element in a video and getting more information or the possibility to buy the item displayed in the video is called "hotspotting" [Ree10; Bee10; Cli12]. The hotspot may move within the video [Ree11]. Its shaping ranges from small icons to the precise outline of an object in the video. Clicking a hotspot pulls up additional information (mainly text and images) and links to external sites [IW06]. The whole presentation is based on a single linear main video. Extended timelines may allow the viewer to jump to a point in time where a hotspot is displayed [Vid12; Wir12]. Additional information is mainly displayed as an overlay over the video. Some commercial players like the VideoClix or ConciseClick player provide one extra area for additional information [Vid12; Cle12]. Clickable videos are mainly used for monetizing products or services in the Internet. A different definition of "clickable video" extending the term to a simple form of "hypervideo" is given by Sengamedu: "A clickable video is referred to as a hypervideo. The clickable regions on a hypervideo are referred to as video hotspots. Hence, hypervideo is based on the premise that regions or objects in a video should be made clickable. [...] [They] can lead to new or further information. Typically, such information is in the form of video, web page, email address, and so on" [Sen09]. With these findings, we define the term "clickable video" as follows:

Definition 2.1 (*Clickable Video*)

> *Clickable videos consist of a linear main video and hotspots. A click on these hotspots makes available additional information, mainly text, images, and links, as an overlay in the video area or on an additional side region. Hotspots can have a different appearance ranging from small icons to outlines of an object in the video. They may move as the element in the video moves.*

2.1.2. Interactive Video

In order to characterize the term "interactive video", we first want to define the term "interactivity". "In a very general definition, interactivity is a sequence of action and reaction" [Dij05, p. 8] according to van Dijk. This definition is formulated in a very general way. Crawford refines the term a little more as "a cyclic process between two or more active agents in which each agent alternately listens, thinks, and speaks - a conversation of sorts" [Cra12, Chapter 2]. He precises this statement for a computer which "accepts input, processes input, and outputs results" [Cra12, Chapter 2]. In a more precise definition van Dijk describe four levels of interactivity [Dij05, p. 8]:

[3] `http://longomatch.org/features.php` (accessed April 26, 2014)

- "The most elementary level of interactivity is the possibility of establishing two-sided or multilateral communication." (*space* dimension)
- "The second level of interactivity is the degree of synchronicity." (*time* dimension)
- "The third level of interactivity is the extent of control exercised by the interacting parties. This *behavioural* dimension is defined as the ability of the sender and the receiver to switch roles at any moment. Furthermore, it is about the control over the events in the process of interaction."
- "The fourth and highest level of interactivity is acting and reacting with an understanding of meanings and contexts by all interactors involved." (*mental* dimension)

The technologies and tools described in this work are capable of the first three levels of interactivity. With the clarification of the term "interactivity", we can now analyze the term "interactive video" and how it is used in related work.

"Interactive video is a subset of interactive multimedia and hypermedia technology where the video content defines the timeline of the presentation and is thereby the driving force" [HVL01]. Only basic interactivity is described for interactive video-on-demand services. The interactive functions include play, pause, stop, fast-forward, and fast rewind [ZA03; ZA05; IA01; LC03]. These are extended by different speeds for fast-forward/rewind [Fei+99; Fei+05; LL98; Par83] as well as jumps for- and backwards [Fei+99; Fei+05; LL98]. Some works enhance these functions with reverse playback in different playback speeds [Li+96; LL98]. Extensions in interactivity indicate considerable differences in literature which are discussed hereafter. "The basic idea of interactive video is to provide more complex operations and feedback to users" [CHC08]. The main video is altered. Thereby, "different view angles, or different zoomed-in and slow-motion factors" [Fer+12], "zooming into individual frames" [NT11], "resolution scalability, progressive refinement (or quality scalability), spatial random access, and highly efficient compression" [NT11] are provided. Furthermore, "the user sets the pace, skimming through familiar material and pausing to review new or difficult ideas" [Par83]. Besides modification of the main video, intervention from the user is required and the interactive video reacts on the input. Tests and decisions are used to determine which parts of the video are shown in which order [YKS99; Par83; MD89]. While the contents of a video are not enhanced in most cases, collaborative features are proposed by [MK91; Kim+11; Pan+12] in different extents. A main component of interactive videos is a browsing functionality which enables a user to access a linear video in a non-linear way. After partitioning a video into smaller segments [MD89; CHS07], single scenes can be omitted [CHS07] or jumps to specific parts of the video are possible [Zha+06]. Zhang et al. "allow proactive and random access to video content based on queries or search targets" [Zha+06] in interactive videos. As a result, "users can select or play a segment with minimal search time" [Zha+06]. Mackay and Davenport state that it is possible to increase the level of interactivity "by allowing users to build, annotate, and modify their own environments" [MD89]. An extension with additional information can be found rarely. Cherrett et al. claim that various media formats like PowerPoint slides, graphics, and simulations "increase the intensity of visual and verbal cues" in interactive videos [Che+09c]. "Interactive objects in the video (text, audio, video, image, web) [...] enable customization of content and make detailed information about the objects in the video available" [See10]. "Moving images, still images, computer graphics" are mentioned as interactive elements by Bosco [Bos89]. Taking these descriptions into account, we define the term "interactive video" as follows:

Definition 2.2 (*Interactive Video*)

Interactive video is mainly based on linear videos and rarely considers other types of media. Basic interactive functions are play, pause, stop, fast-forward, and rewind (at different speeds), as well as jumps for- and backwards. These are extended by more complex functions changing either the presentation of the main video or the sequence of scenes based on user interaction. The video is divided into smaller segments which can be accessed directly. Interactive videos may be extended with additional information.

2.1.3. Non-linear Video

Non-linearity in videos is described in two different ways in the literature: as "video libraries, in which users select from a large collection of videos and may be interested in viewing only a small part of the title; and [as] video walk-throughs, in which users can move through an image-mapped representation of a space" [KWW00] (see Figure 2.1). Thereby, selections in video libraries are classified as "non-real-time, non-linear video applications" and walk-throughs are classified as "real-time, non-linear application" [KWW00]. The contents of video libraries are non-linear before playback, but can be watched linearly without further interaction depending on the playback-software.

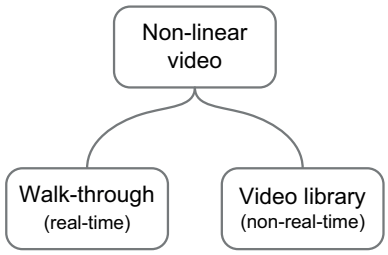


Figure 2.1.: Different types of non-linear video: walk-throughs and video libraries.

Jumps from one scene/part of the presentation to another may be possible [Car+08; YYL96]. The content can be personalized with regard to the viewers preferences [Car+08; ZEV07]. Video walk-throughs are based on a tree or graph pattern and are thus more structured than video libraries. These structures allow the implementation of "different endings depending on the user interactions taking place during story consumption" [Spa+06]. Parallel sequences of frames allow the selection of a branch dynamically during playback [ZEV07]. A non-linear video presentation may be defined as a directed acyclic graph consisting of video- and control nodes linked by edges which define predecessor-successor relationships originating from one start node [KWW00]. Libraries and walk-throughs have in common, that there is "more than one typical ordering of video frames delivered to clients" [KWW96] which allows "multiple possible playback paths differing in the media portions they include and/or their ordering" [Car+08]. The presentation of one final flow is composed on the basis of user interaction [Got06; Hau08]. Furthermore, the term "non-linear video" is mainly subject to the areas of video streaming and broadcasting. No detailed descriptions on how the user selects a scene are given. Additional information in form of images, texts, audio files, or videos can rarely be found [Hau08; KWW00]. In consideration of the described characteristics, we define "non-linear video" as follows:

Definition 2.3 (*Non-linear Video*)

Non-linear videos consist of a set of scenes or video sequences, which are arranged in a library or as a walk-through in a graph structure. Selection elements are provided to either create a video selection from the library or to select a path in the graph, leading to an individualized presentation flow. Additional information and functions to control the reception of the playback are not part of non-linear videos.

2.1.4. Multimedia Presentation

Most of the definitions of the term "multimedia presentation" have three elements in common: static and continuous media, temporal relationships (sometimes leading to synchronization issues), and spatial relationships. It can be noted that videos, audio files, text, and images are part of many multimedia presentations[4]. Other types of media are only mentioned rarely or in multimedia presentations for special purposes like tables and charts [BO98], 3D graphics/models [BGVO05a; HFP99], (presentation) slides [DTL06; FR09; HW98], user interaction buttons [LSIR02], animations [Abd08; Lee+99; NKL09; Sap02; Shi98], and 360° omni-directional camera surround views [HFP99]. Synchronization issues and temporal relationships[5] are major problems in multimedia presentations, while spatial relationships and the layout are least considered[6]. Semantic relationships [Ber+05], navigational dimensions [BGVO05b], hyperlinks [LSIR02], and alternative behavior specifications [Mou00; Mou02] are considered rarely. User interactions[7] mostly depend on the build time of the presentation. "A multimedia presentation whose content is predefined before rendering is called a static multimedia presentation. A multimedia presentation whose content is composed dynamically during runtime is called a dynamic multimedia presentation" [KHM08]. Static multimedia presentations allow only VCR-like interactions while dynamic multimedia presentations enable the user to chose certain contents. Dynamic multimedia presentations are arranged either in a tree [Ass99] or a graph structure [LO98; Lee+99; Tsi+06]. We define the term "multimedia presentation" following Nimmagadda, Kumar, and Lu as follows:

Definition 2.4 (*Multimedia Presentation*)

"Multimedia presentations are collections of different media files [...] like text, images, videos, and animations with different resolutions, durations, and start-times. [...] The layout of multimedia presentations is defined by the locations and the start times of the objects" [NKL09]. Pre-rendered static multimedia presentations allow VCR-like user behavior while dynamic multimedia presentations feature additional navigational structures.

[4] [Abd08; AL95; AT03; BO98; BFS00; BGVO05a; BHL92; Bor+96; CGS04; DTL06; Emi+02; Emi+05; Hak09; HFP99; HW98; KU99; LSIR02; MB02; NKL09; Pra00; Sap02; Shi98; Tsi+06]

[5] [Abd08; AL95; ABM07; AD05; AZ01a; AZ01b; BO98; BFS00; Ber+05; BGVO05b; BGVO05a; BHL92; CG03; CGS04; CS97; Emi+02; Emi+05; GZ98; Hak09; HW98; LSIR02; LO98; Lee+99; MB02; Mou00; Mou02; NKL09; Sap02; Shi+99; Sht08; Tsi+06; TR99; ZJ98; Rou+99; Rut+98]

[6] [AD05; BFS00; Ber+05; BGVO05b; BGVO05a; CS97; Hak09; LSIR02; MB02; Mou00; Mou02; NKL09; Shi+99; Tsi+06; Rou+99; Rut+98]

[7] [Abd08; Ada+00; ABM07; Ber+05; CG03; Che+02; Chu+95; CS97; DTL06; HW98; KHM08; LO98; Lee+99; LL98; Sap02]

2.1.5. Hypervideo

Hypervideos are found in different forms in the literature. They either provide non-linear navigation between scenes or they consist of linear videos with additional information. Chambel differentiates between "(1) homogeneous hypervideo, where video is the only medium involved [...] that can be navigated by the user, and (2) heterogeneous hypervideo that integrates other media, providing further and related information to the video" [CZF04] (based on [ZSB02], [CCG01], and [CGa02]). The third form is a hybrid of homogenous and heterogeneous hypervideo. Hyperlinks (to scenes or additional information) are usually represented by hotspots or sensible regions which depend on space and time in the main video [SBS97; BCF02; FB04; GCD02; MD07; SGW03c; SZF05].

[BF05], [CC99], and [FB04] describe linear heterogeneous hypervideos with additional information like text, images, audio files, animations, and other videos. Finke and Balfanz [FB04] furthermore name jumps in the linear main video. The structure of the homogeneous part of hypervideos (or homogeneous hypervideos) varies. Some authors describe links (hyperlinks) between scenes or videos [SGW03c; LB06; HH06; Hun97; MD07; PJT06; SBS96; Sei11]. Sawhney, Balcom, and Smith [SBS96] illustrate resulting graph structures. Tiellet et al. [Tie$^+$10] detail the navigation in videos which can be "embedded in the video and leading to other moments in the video, or somewhere else in the hypermedia space" [Tie$^+$10]. Furthermore, links can "exist outside the video, e.g. a text page or an index, but have specific moments of the video as destination" [Tie$^+$10]. Tiellet et al. extend the navigational features with "searching and indexing, and real-time annotation" [Tie$^+$10] which provides further navigation to the viewer. A more restricted kind of homogeneous hypervideo is called *detail-on-demand hypervideo*. It supports only one link to jump to additional (explanatory) videos at a given time and returns back to the main video automatically [Doh$^+$03; GSW03; Gir$^+$04; SGW03c]. A combination of homogeneous and heterogeneous hypervideo is the main form found in literature [CGa99; CCN11; HH06; MD07; NC10; PJT06; SAP11; SZF05; Tie$^+$10]. Enhanced features and interactive elements are video-based previews [MD07], implicit spatial semantics (different narrative sequences depending on the time the user interacts) [Muj$^+$12], and communication functions [FB04]. A different point of view is described by Aubert and Prie, who define hypervideo "as views on audiovisual [sic] documents associated with an annotation structure" [AP05]. Closely following the definition of Stahl, Zahn, and Finke [SZF05], we define the term "hypervideo" as follows:

Definition 2.5 (*Hypervideo*)

Hypervideo is defined as video based hypermedia that combines non-linear video structuring and dynamic information presentations. Video information is linked with different kinds of additional information (like texts, pictures, audio files, or further videos). Users can mouse-click on sensitive regions (having spatial and temporal characteristics) within the videos to access the additional information (heterogeneous hypervideo) or jump to other scenes (homogeneous hypervideo). Hyperlinks build a graph between main video scenes and additional information.

2.1.6. Annotated Interactive Non-linear Video

The form of video this work deals with is "annotated interactive non-linear video". These are a mixture of elements of non-linear videos, interactive videos and annotations. They can

be seen as a special and restricted form of hypervideo, which follows a defined structure. Video scenes are linked in a scene graph. Each scene may have one or more annotations which are displayed in the video area or in side areas. While several definitions could be found for "interactive video" and "non-linear video", a combination of these terms can be found rarely. Schneider, Braun, and Habinger describe digital storytelling as interactive and non-linear, thereby "temporal points and the sequence of story elements are not predefined" [SBH03]. Hausenblas describes the creation of non-linear, interactive media. He describes a "story world" as non-linear, the story depends on the user's interactions [Hau08]. Robberecht discusses interactivity and non-linearity in learning materials [Rob07], but does not address videos as the main medium. We define the term "annotated interactive non-linear video" as follows (extended from [Ham06]):

Definition 2.6 (*Annotated Interactive Non-linear Video*)

An annotated interactive non-linear video is a digitally enriched form of video materials arranged for an overall concept. It presents additional information like images, texts, audio files, videos, and links to web pages beyond the original content, which are displayed and hidden at specific points in time. Furthermore, it offers new forms of influence and navigation (selection menus, search-function, table of contents) in the video and additional contents. Thereby user interaction may be mandatory or optional.

2.1.7. Overview and Summary

The different types of enriched video are described and defined for the context of this work in this section. Their features are enlisted and summarized in Figure 2.2. As a summary it can be noted that interactive video and non-linear video show no commonalities except the circumstance that both are based on main videos. Hypervideo and annotated interactive non-linear video are very similar, the latter unites all characteristics of non-linear video and some of clickable and interactive video. Clickable and interactive videos are both based on linear videos and contain additional information, but interactive video is more advanced than clickable video. In contrast to interactive and clickable video, videos scenes which can be linked to a non-linear structure are used in non-linear videos, annotated interactive non-linear videos, and hypervideos. Multimedia presentations have a unique position, because they are not based on video. Another term which can be read occasionally is "rich media application". According to Lighthouse Websites, LLC, a "Rich Media Application, such as a Flash application, is visually pleasing, attention grabbing, and tells the story of your website" [Lig13]. This term can be seen as a generic term for the concepts described in this section when they are displayed in a web browser.

2.2. Description Formats, Multimedia Models, and Standards

All types of advanced videos and multimedia presentations require a description of their internal structures and the offered interaction possibilities. These descriptions are mainly file-based and require an underlying model. Two important description languages in this area are SMIL and NCL, which are described in detail hereafter. Descriptive XML standards like MPEG-7 [ISO09], Dublincore [Dub12], MXF [Fer10], P/Meta [EBU11], or TV Anytime [ETS05] are not suitable for the definition of annotated interactive non-linear video (for a detailed

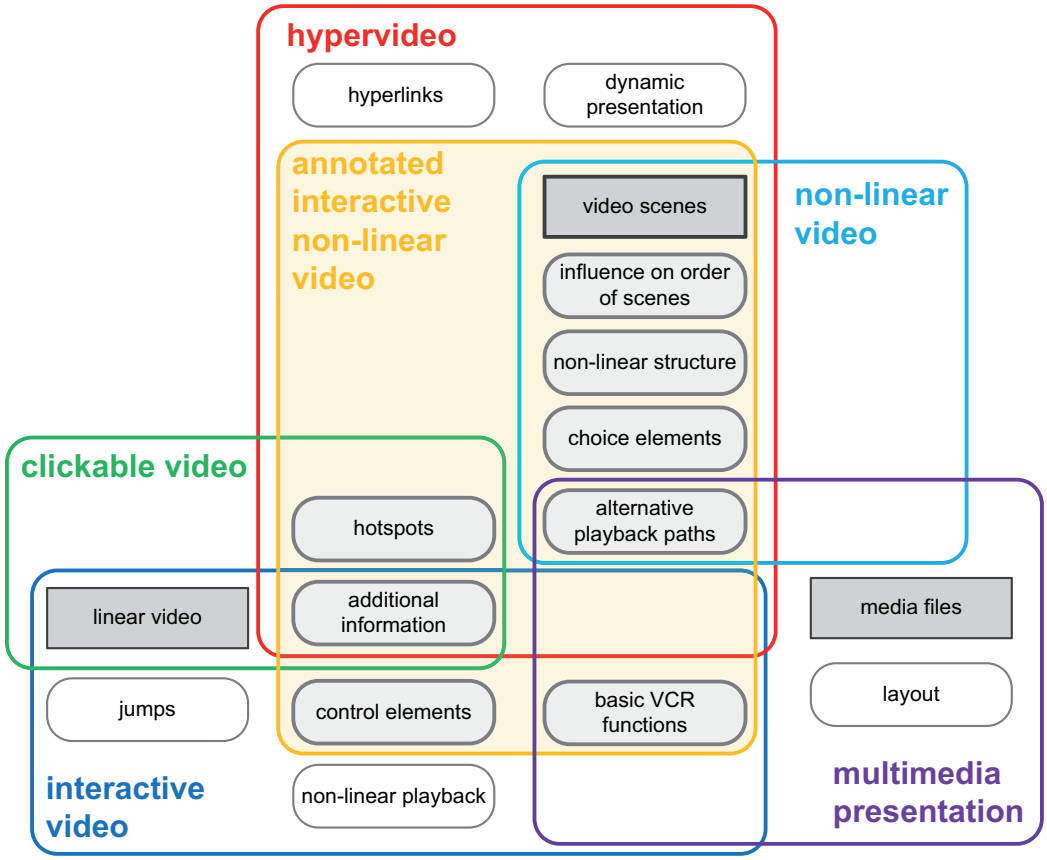


Figure 2.2.: Overview over the different kinds of video: clickable video, interactive video, non-linear video, multimedia presentation, hypervideo, and annotated interactive non-linear video.

overview see [Vic11]). It is not possible to link elements to parallel storylines, to create cycles in the storyline, or to synchronize elements using them. The Sharable Content Object Reference Model (SCORM) [Adv09], a standard from the area of e-learning is designed for Learning Management Systems (LMS). The latest version of the standard, SCORM 2004 4th Edition, is capable of creating complex branching structures, but lacks of metadata definitions to describe the intra-scene interactivity. The following subsection gives a detailed description of models for hypervideos due to their similarities with annotated interactive non-linear videos. Other models and description languages are described briefly. One focus in the area of multimedia presentations is on the compliance of timing constraints which are mainly used in multimedia presentations. Annotated interactive non-linear videos have some commonalities with hypervideos and multimedia presentations but provide extended forms of navigation.

Requirements for an XML data structure for annotated interactive non-linear videos are the feasibility to define temporal and spatial relationships between videos and annotations. Elements needed at a fork in the video flow, like button panels or quizzes need to be defined in the structure as well. Constructs for a table of contents and a keyword reference list are required to implement the extended navigation characteristic for annotated interactive non-linear videos. The structure has to be extensible in case of new ways of interaction that should be mapped into the model. An event-based timing model is preferred to a structured timing model. Timing issues should be kept as local as possible.

2.2.1. Models and Languages for Interactive Multimedia Presentations

The two most important languages from the area of interactive multimedia standards are the Synchronized Multimedia Integration Language (SMIL) and the Nested Context Language (NCL). Both could be used (with some extensions) to describe annotated interactive non-linear videos, but are not designed for this purpose. Another combination of standards - HTML5 [W3C13c], CSS [W3C13a], and SVG [W3C13d] - can be used to write documents or websites which are then displayed in Web browsers. The elements of HTML allow the definition of metadata, to enable scripting (for example with JavaScript [Moz13b]), create hyperlinks to other websites, and to define section and group contents. Contents like images, videos, and audio files can be embedded in the code. Furthermore, it is possible to create tables and implement forms for data submission. These elements can be used to create parts of multimedia presentations for the Web with may then be hyperlinked with each other. The usage of JavaScript in combination with these three standards allows the creation of players which may read XML [W3C03] or JSON [Ecm13] files which describe interaction and non-linearity of a video.

2.2.1.1. SMIL

The Synchronized Multimedia Integration Language (SMIL) is a standard for interactive multimedia presentations released by the World Wide Web Consortium (W3C). Design goals of SMIL were to define "an XML-based language that allows authors to write interactive multimedia presentations. Using SMIL 3.0, an author may describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen. [Furthermore, it should allow] reusing of SMIL 3.0 syntax and semantics in other XML-based languages, in particular those who need to represent timing and synchronization" [W3C12]. Used media files are images, text, audio files, video, animation, and text streams which are linked to an internal graph/tree structure. Navigation is possible in a presentation, but not in single continuous media files. Furthermore, it is possible to define hotspots for navigation or to display additional information. With the usage of the elements and attributes from the timing modules, "time can be integrated into any XML language" [BR08, p. 117]. It is possible to define start and end time, duration, persistence, repetition, and accuracy of objects and relation between those objects [BR08, p. 117]. The layout of a presentation is defined by the "relative placement of (multiple) media objects", but SMIL does not involve the internal formatting of media objects [BR08, p. 149]. SMIL is based on CMIF [BRL91] and the AHM [HBR94].

The final version of this standard is the SMIL 3.0 Recommendation, which was published on December 01, 2008 [W3C12]. Previous versions of this standard were SMIL 1.0 released in 1998, SMIL 2.0 released in 2001, and SMIL 2.1 released in 2005 [BR08]. SMIL 3.0 consists of 12 major functional areas of elements and attributes (Animation, Metainformation, Content Control, Structure, Layout, Timing and Synchronization, Linking, Time Manipulations, Media Objects, Transition Effects, smilState, and smilText) described as a DTD. The "Timing and Synchronization" part is the most important [BR08]. Furthermore, five profiles are built which use the enlisted elements and attributes, namely the SMIL 3.0 Language Profile, the SMIL 3.0 Unified Mobile Profile, the SMIL 3.0 DAISY Profile, the SMIL 3.0 Tiny Profile, and the SMIL 3.0 smilText Profile [W3C12]. These profiles may limit the elements and attributes of the standard or extend it with functionality from other XML languages [BR08].

Related work dealing with the SMIL format either checks the temporal constraints of a SMIL file for their consistency or extends SMIL. Chang proposes an "intelligent methodology to diagnose the temporal consistency of [a] SMIL document that supports the presence of multiple distributed multimedia objects, as well as human-computer interaction" [Cha05]. Therefor, he develops "a temporal algebra system as multimedia synchronization model to unify media presentation time and interaction event" [Cha05]. Elias et al. extract the temporal layout of a SMIL file to generate a dynamic petri net which is then used "(i) to serve as a guide for the run-time component and (ii) to perform verification of the set of specifications" [Eli+09]. Chung and Pereira transform the SMIL document into a "Timed Petri Net (TPN)" [CP05] to be able to schedule the elements of a SMIL presentation accordingly. Sampaio, Santos, and Courtias introduce a method for the semantic verification of SMIL documents in [SSC00]. Using this method, they identify erroneous interpretations in currently available SMIL players. Gaggi and Bossi introduce a tool for the automatic verification of SMIL documents. They use "formal semantics defining the temporal aspects of SMIL elements by means of a set of inference rules" [GB11].

Extensions for SMIL can be found in different areas. Hu and Feijs describe "IPML, a markup language that extends SMIL for distributed settings" [HF06]. SMIL concepts are brought into HTML and web browsers by HTML+TIME [SYS98]. Hereupon is XHTML+SMIL based. It "defines a set of XHTML abstract modules that support a subset of the SMIL 2.0 specification. It includes functionality from SMIL 2.0 modules providing support for animation, content control, media objects, timing and synchronization, and transition effects. The profile also integrates SMIL 2.0 features directly with XHTML and CSS, describing how SMIL can be used to manipulate XHTML and CSS features. Additional semantics are defined for some XHTML elements and CSS properties" [W3C02]. Limsee3 tries to simplify the authoring process of SMIL files by providing templates for certain purposes. Thereby it integrates "homogeneously logical, time and spatial structures. Templates are defined as constraints on these structures" [DR06; MRLD08]. Terashima et al. propose a language called QOS-SMIL which adds QoS to a subclass of SMIL 1.0 using real-time LOTOS [Ter+00].

Vaisenberg, Jain, and Mehrotra [VJM09] introduce the SMPL framework which is able to add a table of contents, a search function, and a bookmark function to SMIL presentations. Thereby, a semantic layer is added to SMIL presentations. Pihkala and Vuorimaa describe "nine methods to extend SMIL for multimedia applications" (like for example multimedia consoles) in [PV06]. Thereby, SMIL 2.0 is extended with "location information, tactile output, forms, telephoning, and scripting" [PV06]. A generic, document-oriented way to publish multimedia documents on the web using HTML5, CSS, and SMIL Timesheets is called Timesheets.js and presented by Cazenave, Quint, and Roisin in [CQR11]. The combination of the different standards allows to merge logical and temporal structures. Additional libraries provide a table of contents and other forms of navigation.

The Narrative Structure Language (NSL) which is used together with SMIL is proposed by Ursu et al. in [Urs+07a; Urs+07b; Urs+08]. NSL can be used to achieve a variation in pre-recorded materials "by selecting and rearranging atomic elements of content into individual narrations" [Urs+08]. The basic elements in this language are "Atomic Narrative Objects (ANO)" [Urs+08]. Interactions for and links between ANOs can be defined whereby a scene graph is built. Different types of so called "selection groups" (comparable to our selection control element) can be defined. Selection criteria for ANOs (or paths in the graph) can be specified with Boolean expressions. Furthermore, different types of variables are stored. These can be accessed by the language. The NSL uses its own computational language syntax which

makes a direct translation into XML impossible. Both the forward button and the conditional selection element can be expressed using the NSL.

Several other extensions for different versions of SMIL exist. Some extensions of one version of SMIL became part of the subsequent version of the standard. Bulterman examines SMIL 2.0 for document-related requirements of interactive peer-level annotations in [Bul03]. An extension to XLink 1.0 called XConnector is proposed by Muchaluat-Saade, Rodrigues, and Soares in [MSRS02]. Reaction to user inputs of different forms is integrated into XML documents and evaluated with real time programming by King, Schmitz, and Thompson in [KST04]. Both extensions are applicable to SMIL 2.0 documents. An extension for SMIL 2.1 called SMIL State is proposed by Jansen and Bulterman in [JB08] and [JB09]. It allows one to add variables to a multimedia presentation enabling dynamic adaptation to user interactions. SMIL State became part of SMIL 3.0. A temporal editing model for SMIL 3.0 is described by Jansen, Cesar, and Bulterman in [JCB10]. Thereby, different forms of document transformations are analyzed.

Critical reflection: Many of the tasks related to annotated interactive non-linear video can be implemented with the Synchronized Multimedia Integration Language 3.0 (SMIL 3.0). "Using SMIL, an author may describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen" [W3C12]. All basic navigation issues and the attachment of annotations to scenes can be implemented straightforward. However, a search functionality for keywords is not provided in SMIL. The `metadata` element of SMIL 3.0 could be used to add keywords to scenes and annotations. This non intended use of the element has to be implemented in the search function in players. No structure for a table of contents is provided by SMIL. It is possible to arrange `text` elements in form of a static tree-based structure. But by this mechanism, branches of the tree cannot get collapsed because no basic function is implemented in SMIL players therefor. Quizzes may be used as a decision module at a fork in the e-learning scenario. One page of a quiz can be modeled with `text` elements similar to the table of contents, the current score can be saved in a `state` element for later usage. The player has to implement this functionality accordingly. The SMPL framework for SMIL may provide ideas for the implementation of a table of contents and a search function. A more detailed analysis of the suitability of SMIL for annotated interactive non-linear videos can be found in [Ber12a].

2.2.1.2. NCL

The Nested Context Language (NCL) is a declarative XML-based language for hypermedia document authoring designed at the "TeleMidia Lab - PUC-Rio" [Tel11]. It is standardized as "H.761: Nested context language (NCL) and Ginga-NCL" [Int11]. Being designed as a hypermedia document specification for the Web, its main field of application are DTV systems [Tel11]. "As NCL has a stricter separation between content and structure, NCL does not define any media itself. Instead, it defines the glue that holds media together in multimedia presentations. [A] NCL document only defines how media objects are structured and related, in time and space" [Tel11]. Variable and state handling in NCL is described and discussed by Soares et al. [Soa+10]. It describes the temporal behavior of a multimedia presentation and the layout of elements on different end user devices. Furthermore, user interaction with single objects can be defined as well as the activation of alternative parts of a presentation [Int11]. Media files which can be linked with each other are images (JPEG, PNG, etc.), video (MPEG,

MOV, etc.), audio files (MP3, WMA, etc.), and text perceptual objects (TXT, PDF, etc.). Furthermore, objects with imperative code content (LUA code, etc.) and objects with declarative code content (HTML, LIME, SVG, MHEG, nested NCL applications, etc.), including other NCL embedded objects [Tel11; Int11] can be added. NCL is based on the Nested Context Model (NCM) [Cas+91; SR05] and inherits modules from SMIL [Sil+04].

The current version of this standard is version 3.0. Previous versions of this language are NCL 1.0 which was defined as a DTD. The second version, NCL 2.0, was defined in a modular way using XML Schema. According to that, a combination of single modules in language profiles was possible [Tel11]. NCL 2.0 contained 21 modules from eleven functional areas [Sil+04]. Versions 2.1, 2.2, 2.3, and 2.4 refined previous versions and introduced new modules [Tel11]. NCL 3.0 specifies attribute values and introduces new functions named "Key Navigation" and "Animation". Furthermore, "NCL 3.0 made [in-]depth modifications on the Composite-Node Template functionality. NCL 3.0 also reviewed the hypermedia connector specification in order to have a more concise notation. Relationships among NCL imperative objects and other NCL objects are also refined in NCL 3.0, as well as the behavior of NCL imperative object players" [Tel11]. NCL 3.0 contains 29 modules and four different predefined profiles. NCL 4.0 is work in progress[8].

Critical reflection: Just as for SMIL, NCL does not provide native structures to define a table of contents or a list of keywords with associated scenes or annotations. Furthermore, the construction of quizzes is not possible, as well.

2.2.2. Other Multimedia Presentation Models and Languages

Other **multimedia presentation/document** models and languages, as those previously described, are described by Adali, Sapino, and Subrahmanian [ASS99; ASS00], Adiba and Zechinelli-Martini [AZM99], Assimakopoulos [Ass99], Deng et al. [Den+02a], and Scherp and Boll [SB05]. Further models are ZYX [BKW99; BKW00; BK01], the Layered Multimedia Data Model (LMDM) [SW94], Madeus [LSI96], and MPGS [BFS00]. Interchange formats are the CWI Multimedia Interchange Format (CMIF) [BRL91] and the Procedural Markup Language (PML) [Ram+99]. Both, models and formats are described for PREMO (Presentation Environment for Multimedia Objects) [HRL96a; HRL96b] and XiMPF: eXtensible Interactive Multimedia Presentation Format [VA+04]. Models and formats commonly consist of a temporal and a spatial model/description defining when and where media elements are displayed. Media elements are in general videos, audio files, images, and texts. Furthermore PREMO and PML allow the usage of animated graphics. Jumps on the timeline can be specified in LMDM and in the approach described by Scherp and Boll.

Models and languages for **hypermedia applications** are HyTime [Gol91; NKN91; Erf93], the Amsterdam Hypermedia Model (AHM) [HBR94; HB97; HWB97]), MHEG-5 [Ech+98], and the model described by Celentano and Gaggi [CG00; GC05]. These models and languages are more general in describing relations between media objects which define a graph structure. Usable media files are video, audio files, images, and text. User interaction varies from jumps in a timeline to jumps in a graph structure enabling the viewer to get additional information of some kind.

[8] `http://www.telemidia.puc-rio.br/?q=pt-br/projetoNCL40` (accessed April 26, 2014)

Reference models and frameworks try to divide the tasks of describing and presenting a multimedia presentation into several layers. Bordegoni et al. describe a "standard reference model [which] consists of several layers referring to the particular subtasks which occur in multimedia presentation generation" [Bor+96; Bor+97]. Shih describes three implementations of this model in [Shi97]. The AMF (Amsterdam Multimedia Framework) "provides an explicit partitioning of control concerns across components in a network infrastructure" [Bul93].

A common problem in many of the description languages and models (especially in interval-based models) for multimedia presentation is the **temporal synchronization** of the elements, which may occur in different temporal relationships as described by Allen in [All83]. Different approaches are published which try to overcome this problem. Cruz and Mahalley propose an efficient approach to determine "whether the presentation is synchronized [...], or amenable to synchronization [...], or impossible to be synchronized [...]" [CM99]. Prabhakaran adapts structure, content, and view based on resource availability, access constraints, and user preferences [Pra00]. Hakkoymaz, Kraft, and Ozsoyoglu define inclusion, exclusion, and presentation organization constraints to create multi-stream presentations in an automated way [HKO99]. Several approaches are used to analyze and synchronize media elements. Huang and Wang use a "dynamic extended finite-state machine (DEFSM) model" [HW98], Shih et al. use the Z notation [Shi+99], Little and Ghafoor use "Timed Petri Nets and the logic of temporal intervals" [LG90a; LG90b], Tan and Guan use dynamic Petri Nets (DPN) [TG05], and Shih uses a collection of Petri nets [Shi98]. Furthermore, different types of synchronization models and languages are proposed. The Firefly multimedia document system described by Buchanan and Zellweger combines compile time and runtime temporal formatting [BZ05]. PROMELA/SPIN described by Aygün and Zhang contains a "synchronization model [which] has receivers, controllers and actors to handle events, condition expression and actor expression, respectively" [AZ02]. A "declarative synchronization definition language" is used by Bailey et al. in Nsync [Bai+98]. Hakkoymaz uses an "event point model" to describe the temporal layout of segments and their play out order [Hak09]. Meira and Moura present an "object-oriented formal specification language" to solve the synchronization problem [MM94]. Presti, Bert, and Duda use a language based on Temporal Algebraic Operators [PBD02]. Schnepf, Konstan, and Du describe a model for specifying coarse synchronization for flexible presentations called FLIPS [SKD96]. A detailed overview of this problem and a comparison of possible solutions are described by Blakowski and Steinmetz in [BS96].

Synchronization issues get even more complicated, when **user interaction** is allowed. Wahl, Wirag, and Rothermel name temporal interaction forms and their temporal dependencies and suggest an integrated model for time and interaction [WWR95]. Existing languages can be extended by the properties described by Bes and Roisin namely "priorities, more abstract properties and fall-back positions" [BR02]. Keramane and Duda extend basic media segments with "executable code, live feeds, and links" [KD97]. Thereby they take user interactions, content-sensitivity, and new sources of multimedia data into account while providing a support for sharing and reuse [KD97]. A more detailed overview of all multimedia models, languages, and standards can be found in Appendix B on page 213.

Critical reflection: Models and languages for multimedia presentations focus on the temporal and spatial arrangement of media objects. Interaction with these objects is rarely possible. Navigation in these presentations is commonly on a timeline. A table of contents or the selection of parts of the presentation are not provided. Hypermedia applications are more general in their definitions. Reference models give advice on how tasks in displaying multimedia pre-

sentations can be divided into different layers. A main problem in multimedia presentations is the temporal synchronization of media elements in interval-based models. Synchronization is even more difficult if user interaction is taken into account. The temporal model of annotated interactive non-linear videos is event-based to avoid these problems, but ideas for spatial layout and the division of tasks into layers can be used and adapted from related work.

2.2.3. Models and Languages for Hypervideos

Four different models were identified for the description of hypervideos. All models have video as a main medium. Annotations are mainly images, text, audio files, and videos. The video scenes are linked to a graph structure by the definition of hyperlinks. The models can be described as follows:

Chambel and Guimaraes [CGa02] describe a "hypervideo model [which] is based on the hypermedia model of the Web, extended with additional concepts required to support the temporal dimension of hyperlinking in dynamic media, such as video" [CGa02]. The main medium in this model are videos which are enriched with images and text. The media are linked to [a] graph structure by hyperlinks. Different types of links like "multi-links, dynamic links, synchronized links" [CGa02] as well as a table of contents and various maps are used to navigate in the hypervideo. "Link anchors can be spatially scattered in the pages and images, allowing for a more fine grained addressing of links origin and destination" [CGa02]. Jumps to points on the timeline can be defined in a video index. Temporal links which are established for a time interval are dependent on time conditions. Spatial links depend on space conditions and make it possible to establish links from certain regions of the video. The language used for hypervideo construction is called HTIMEL.

Generalized HyperVideo System (GHVS) model [Hun97]: GHVS can be used to specify hyperlinks between frames. Furthermore, it meets "basic goals like physical data independence, the ability to compose arbitrarily complex presentations of hypervideos, completeness in expressibility, and simplicity" [Hun97]. A graph consisting of video scenes is defined by video-to-video hyperlinks. Rectangled hotspots allow the definition of jumps to other frames, between scenes, and to audio files and images. The defined language in this work is called GHVS language and it is based on the "Generalized HyperVideo System (GHVS) model" which in turn is based on the PRES model [WKD96].

Logical Hypervideo Data Model (LHVDM) [JE98]: "In addition to multilevel video abstractions, the model is capable of representing video entities that users are interested in (defined as hot objects) and their semantic associations with other logical video abstractions, including hot objects themselves" [JE98]. Links between videos define a graph structure. Furthermore, it is possible to define links in videos to jump to certain frames. The order of scenes cannot be defined before playback. Contents shown with the videos, like images and audio files are extracted from the main video by creating images out of frames or audio files by saving the soundtrack. Temporal information describe the time intervals during which an object is activated as link (hot object). The object has a certain spatial information thereby. Furthermore, spatial relations between hot objects exist. A video query language is defined for the LHVDM.

Component-based Hypervideo Model (CHM) [SAP11]: The CHM is a "high level representation of hypervideos that intends to provide a general and dedicated hypervideo data model" [SAP11]. This model consists of a spatial, a temporal, and an event-based model. The main medium is video. Videos are linked and extended with text, video, audio files, and rich text. Jumps to points on a timeline, in a map, in a history, or to links associated with a table of contents are possible. The model provides "high level components such as data readers, enrichment content viewers, video players, hotspots, timelines, maps and tables of contents" [SAP11]. A timeline-based model with virtual time references is used. The spatial model is derived from SMIL.

Critical reflection: The models described in this subsection provide some of the desired functions needed in annotated interactive non-linear videos. All models provide timing and spacial settings in the following way: Basic constructs to describe the position of an object are available, but it is not possible to define moving areas in some models. The timing models vary in their way of synchronizing single elements. Two of the models provide a table of contents, but no structures to define keywords for a search function. Both of these functions need a linking between a label (either an entry in the table of contents or a keyword) and the jump destination (a scene or annotation). The model described by Chambel and Guimaraes does not provide the impression of an over-all video, because the linking is realized between websites with embedded videos and not in a single video player which loads different video files. The Generalized HyperVideo System (GHVS) model is a basic and relatively static model which does not provide a table of contents or keywords in addition. No annotations are used in the LHVDM which focalizes hotspots and linking. None of the models was transferred into a usable (XML) language. The shortcomings of the models can be summarized as follows [MK12]:

- No currently available XML format for the definition of annotated interactive non-linear videos is on the one hand restrictive enough to be read flawlessly by players and on the other hand extensive enough to define such videos for a wide variety of scenarios.
- No event-based model with a simple time synchronization is available, which is preferred because of a possibly high level of interactivity.
- No DTDs or XSDs are available for the models described in this subsection.
- None of the models provides a keyword search.

The described shortcomings lead to the following consequences:

- Existing formats need to be restricted on the one hand and extended on the other hand to be able to implement all requirements for annotated interactive non-linear videos.
- An event-based model with a simple time synchronization is preferred to handle the possibly high level of interactivity.
- An XSD is preferred to a DTD, because it is more restrictive. Less errors in the resulting XML file are possible as a consequence.
- Keyword search and table of contents should be definable by the XML format natively for a seamless integration of the whole concept.

2.2.4. Summary

This section is a review on the models and languages which can be used to describe the varying forms of video presentations. Models from the different areas show shortcomings which make them unusable for annotated interactive non-linear videos. Some models and languages could be used with extensions for certain elements and functions on the one hand and with restrictions on functions on the other hand. They are not suitable for the form of video we are dealing with in their current versions. The best-known standard for multimedia presentations, SMIL, lacks of important functions and uses a time-based instead of an event-based timing model which makes it difficult to use for annotated interactive non-linear videos.

Figure 2.3 shows the chronology of the publication/standardization of description formats, standards, and models (the first publication is marked thereby). First publications are from 1991. Especially CMIF, AHM, and Madeus form the basis for later works and the SMIL standard. It furthermore should be noted, that from 2006 on only a few new developments could be found.

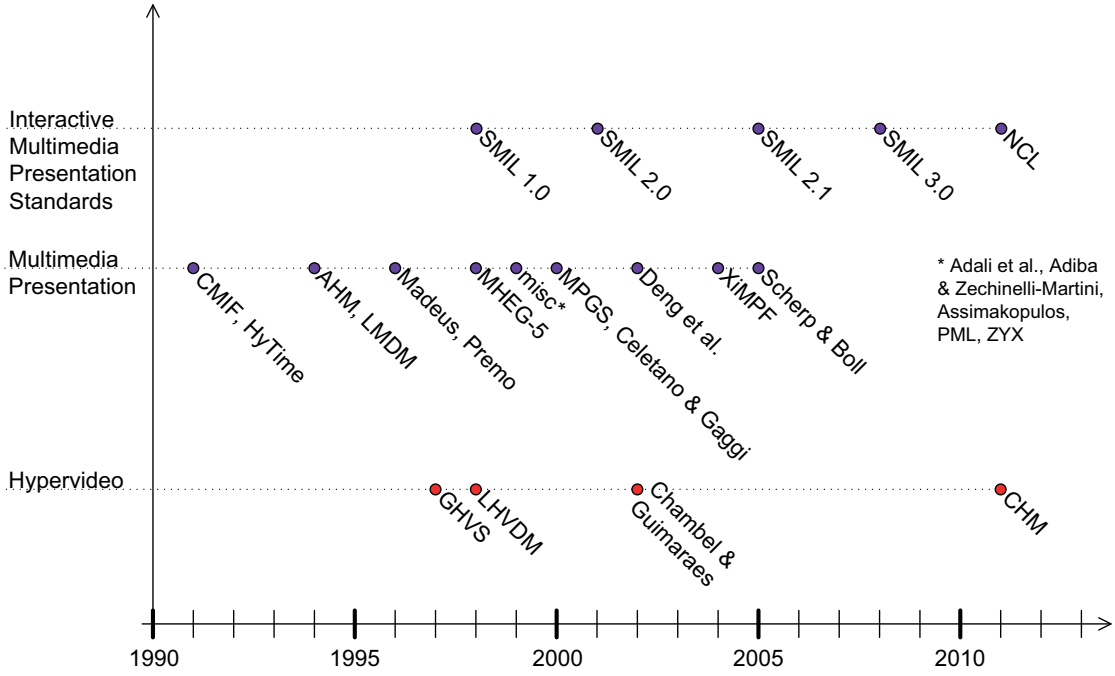


Figure 2.3.: Chronology of the publication/standardization of description formats, standards, and models.

(Parts of this section (2.2 Description Formats, Multimedia Models, and Standards) were taken and adapted from our previous work [MK12].)

2.3. Evaluation of Existing Authoring Tools

The authoring process of interactive and non-linear media is more complicated than the process for traditional linear video. Back in 1989, Fox demanded that "efficient tools and environments for authoring and editing of interactive multimedia programs must be developed" [Fox89]. At about the same time, Mackay and Davenport as well as Hodges, Sasnett, and Ackerman described multimedia tools developed by the MIT [MD89; HSA89]. Those tools offer functionalities like recombination of video segments, synchronization of media, live annotation, video analysis, and sharing of interactive video data. In 1991, Davenport, Smith, and Pincever described a structured annotation model called "*Stratification*" which describes shots [DSP91]. In the past twenty years many efforts have been made to enable "automatic re-structuring, indexing and cataloging [of] video content" or to provide "advanced interaction features for audio-video editing, playing, searching and navigation" [Ham06]. Bulterman and Hardman describe "issues that need to be addressed by an authoring environment" for multimedia presentations in [BH05]. They identify "seven classes of authoring problems", namely the definition and realization of media assets, synchronization composition, spatial layout, asynchronous events, adjunct/replacement content, performance analysis, and publishing formats [BH05]. Furthermore, they describe and explain "four different authoring paradigms", which are (please refer to [BH05] for further reading): structure-based, timeline-based, graph-based, and script-based. Our literature review showed that not all tools deal with all of the seven problems. The authoring paradigms are enhanced by other GUI elements and paradigms to provide the full functionality needed in an authoring tool. The remainder of this section gives an overview of authoring tools for clickable videos, interactive videos, non-linear videos, multimedia presentations, and hypervideos. Thereby the categorization established in Section 2.1 is used, even if single tools are categorized in a different way by the authors of the described works. Characteristics presented in Figure 2.2 may not be implemented in any tool named in this section, because the definitions in Section 2.1 are based on a wider range of related work which might not always have resulted in prototypical implementations. Some of the described tools are a hybrid form of the characterized types of video. They are then described with the group of authoring tools they show the most similarities with. Authoring tools for hypervideos are analyzed in more detail, because of their similarity to annotated interactive non-linear videos. A more detailed overview of the authoring tools described in the remainder of this section can be found in Appendix C on page 223. Tools described in scientific work were not tested for the use with current operating systems. (Parts of this subsection were taken and adapted from our previous work [Mei+12b].)

2.3.1. Other Authoring Tools

Authoring tools for **clickable videos** like Klickable [Kli13], Overlay.TV [Ove10], VideoClix [Vid12], Viddix Beta [VID10] enable authors to add additional information mainly in form of text, images, and links to one linear main video. Klickable and Overlay.TV focus on online shopping and allow to provide shopping carts while Viddix Beta offers a poll and an RSS-feed. Hotspots can be defined in form of rectangles (Klickable, Viddix Beta), as images of an object (Overlay.TV), and as an overlay over a (moving) object in the video (VideoClix). Sensarea, described by Bertolino, is a software "that can be used in a post-production environment and that allows to automatically or semi-automatically perform spatiotemporal segmentation of video objects" [Ber12b]. Masks of objects in the video are produced as output thereby and

can than be used to create clickable videos. A description language or format can be found for neither of the tools. The editing is done in WYSIWYG editors or video previews, both combined with input forms.

Tools for **interactive videos** are Composer [LGaMDRCGS08], HyLive [HKH08], Zodiac [Chi+00], Popcorn Maker [Moz13a], Quicktvpro [Bel12], SeViAnno [Cao+10], 5minMedia VIDEO EVERYWHERE [5mi14], ADIVI Production Kit [Inn11], HyStream System [Bea+02], wireWAX [Wir12], LazyMedia [HL06], ConnectME [NBB13], and the tools described in the work of Chang et al. [CHS07; CHC08], Räck, Seeliger, and Arbanowski [RSA10], and Chen et al. [Che+09a]. These tools either allow to create linear video structures (Chen et al., Zodiac, Popcorn Maker, Composer, and Chang et al.) or the resulting interactive video is based on one single linear video (the 5minMedia platform, Quicktvpro, Adivi, HyStream, SeViAnno, wireWAX, LazyMedia, ConnectME, and HyLive). While no non-linear structure can be created and no influence on the order of scenes before playback is possible due to the characteristics of interactive videos, alternative playback paths (jumps) in the video are definable. Chang et al., Quicktvpro, LazyMedia, and the 5minMedia platform enable the author to create entry points to switch to other video segments. Navigation by annotations can be defined using SeViAnno. Control elements can be defined with the tool of Chang et al. in form of buttons that appear in the video and require user input. In the tool described by Chen et al. one or more interactive elements have to be defined. They urge the user to interact with the video before entering the next scene. A list of scenes of one video can be defined using the 5minMedia platform, LazyMedia, and Quicktvpro. Usable types of additional information are text and images. Links to websites can be created with the tool described by Chang et al., ConnectME, and the 5minMedia platform. The tool described by Chen et al. allows the creation of small gaming units. Popcorn Maker allows the integration of the Google Maps API[9], the Twitter API[10], and other social websites/APIs. Presentations are added to a new video in the HyStream system. Semantic annotations can be placed within the video using SeViAnno. HyLive and Quicktvpro provide the possibility to add voting to a video among other forms of annotation. The definition of Hotspots is possible using Zodiac, Popcorn Maker, the 5minMedia platform, wireWAX, HyLive, Quicktvpro, and ADIVI to invoke additional information (for example by opening a website). The appearance of the hotspots varies while most of them have a fixed position in the video. LazyMedia creates video chapters by clickable preview images of the scene, they furthermore add text and image to a scene. Players for different end-user devices are available in the ConnectME project. No common description language is used. Composer uses NCL 3.0 [Tel11], Chen et al. use SCORM [Adv09], Popcorn Maker uses HTML5 [W3C13b] and JavaScript [Moz13b]. The HyStream System uses RDF [RDF04], Räck, Seeliger, and Arbanowski use the Object Definition Language (ODL) [Cat+00], and SeViAnno uses MPEG-7 [ISO09]. No information on the description language is given for the other works described above. The GUI of most tools uses the timeline-based pattern or some kind of video preview/WYSIWYG-editor, both in combination with different input forms.

Non-linear videos can be created with the Riva Producer Enterprise [mem13], with XIMPEL [Bhi+10], or on the YouTube website with YouTube Video Annotations [You13]. These tools allow to create projects based on videos which are then enhanced by images or text. The creation of hotspots is used to link scenes with each other, which results in the non-linear structure. Alternative playback paths (jumps) cannot be applied due to the definition of non-linear videos. An influence on the order of scenes before playback, as described for video

[9] `https://developers.google.com/maps/` (accessed April 26, 2014)
[10] `https://dev.twitter.com/` (accessed April 26, 2014)

libraries, cannot be defined in any of the described authoring tools (this function might be implemented in players). Neither hyperlinks nor time-based annotations can be defined as in hypervideos. The GUIs of YouTube Video Annotations uses the timeline-based pattern combined with input forms and a video preview. No GUI is available for XIMPLE, the description of the structure is edited in XML in an arbitrary XML editor. No documentation of the GUI was found for Riva Producer Enterprise. Description languages for the created structures use self-defined XML formats.

Tools for the authoring of **multimedia presentations** can be split into two groups: tools for *basic multimedia presentations* with no interactivity and tools for *complex multimedia presentations* which allow user interaction of some kind. Nearly all tools allow the combination of text, images, audio files, and videos - often described as media files or media elements. Basic multimedia presentations can be created with SMIL Builder [BB11], GRiNS [Bul+98], SMILAuthor [YY03]/SMILAuthor2 [YCW04a; YCW04b; YCW08], the Synchronization Editor [BHL92], TYRO [Mac91], MPRES Author [WRR97], Java-Assisted SMIL (JAS) [DTL06], SIMPLE [Mur+06], and the tools described by Sung and Lee [SL05], Villard [Vil01], Deng and Shih [Den+02b], Jokela, Lehikoinen, and Korhonen [JLK08], and Shih et al. [Shi98; Shi+98b; Shi+98a]. These tools neither allow the definition of alternative playback paths (jumps) nor do they provide any kind of navigational elements. Prior tools which were developed before SMIL became W3C Recommendation in 1998, like tools described by Shih et al., Deng and Shih, Villard, in the Synchronization Editor, TYRO, SIMPLE, and MPRES Author use self-defined models or XML-formats. Recent tools like SMIL Builder, GRiNS, SMILAuthor, Java-Assisted SMIL (JAS), and the tools described by Sung and Lee and Jokela, Lehikoinen, and Korhonen use SMIL as description language for the interactivity. Tools for more complex multimedia presentations are for example Eventor [Eun+94], Delaunay MM [CL97], Madeus [Jou+98], CMIFed [Ros+93], Harmony [Fuj+91], MEMORY [KHM08], IMMPS [SD97], MediaTouch [Ech+98], LECTURNITY 4 [imc10], ZEEGA [Zee13], NextSlidePlease [Spi+12], Matchware Mediator 9 [Mat12], and the tools described by Cutts et al. [Cut+09]and Gaggi and Celentano [GC02]. These tools offer different additional functions. Eventor is “composed of three tools: a Temporal Synchronizer, a Spatial Synchronizer, and a User Interaction Builder” [Eun+94]. Delauny MM provides a document generation module which can be configured with layout and constraint specifications. Madeus provides “various kinds of context-dependent navigation: step by step navigation [...], structural navigation [...], and user defined navigation[...]” [Jou+98] in a multimedia presentation. In CMIFed, the presentation is controlled by the manipulation of events and timing constraints. Cutts et al. allow the definition of a table of contents, a search function, and markers on the timeline. Harmony provides the definition of “link semantics” which enable the author to define non-linear structures. Complex multimedia presentations with hyperlinks can be created with the tool described by Gaggi and Celentano. Adaptive multimedia presentations can be created with MEMORY, which also allows the definition of search queries for media documents and navigation in the search results. “Intelligent multimedia presentations” can be created using IMMPS. Therefore, “presentation resources, presentation knowledge, navigation rules, and presentation window layouts” [SD97] have to be defined. MediaTouch enables authors to edit MHEG-5 objects providing a hierarchy, a properties, a layout, and a links editor [Ech+98]. NextSlidePlease uses “a directed graph structure approach for authoring and delivering multimedia presentations” [Spi+12] which are mainly based on presentation slides as media objects. ZEEGA allows the composition of different multimedia elements to one single site. These sites are then brought into a linear order. Commercial tools like Lecturnity and Matchware provide functions to create an extended navigation by using hotspots, buttons,

and links. Used patterns and editors vary greatly depending on the complexity of the resulting presentation and the level of interactivity or the extent of additional functions. Most of the described tools for more complex multimedia presentations do either not describe the description language or use self defined formats and models. MEMORY uses LOM [IEE02] and Matchware Mediator 9 uses a combination of HTML [W3C13c] and JavaScript [Moz13b].

Tools which cannot be assigned to any type of the software described above are high end video editing tools and video annotation tools. High end video editing and web development software like Adobe Creative Suite 6 Production Premium[11], Adobe Director 11.5[12], Microsoft Silverlight[13], or Microsoft Expression[14] is capable of producing all types of video described in Section 2.1. Each video has to be created individually and can be suited to special needs of the desired scenario. Only a limited degree of automation is possible. Video annotation tools like Ambulant Captioner [LGaCB10], ANVIL [Kip01], IBM VideoAnn Annotation Tool [Nap+01], The Choreographer's Notebook [Sin+11], and VideoANT [Aca07] are used to add text annotations of contents to a linear video. Segmentation and indexing of the videos is possible using the annotations afterwards.

Critical reflection: Authoring tools for interactive videos are, like video annotation tools and tools for clickable videos, mainly timeline-based. Several additional editors are needed because of more extensive forms of interaction. One of them is an overlay editor. It allows the author to mark objects in the video mainly in a manual way. The mark can be static or adapted to move with the object in the video with more sophisticated tools. Another editor is needed for a table of contents. Entries have to be created and linked with points in the video. Both editors are needed in our authoring tool. The editor for the table of contents has to be implemented with extended concepts. These should allow an author to assign scenes to nodes in the contents tree, because we are dealing with different scenes, not a whole single video. Authoring tools for multimedia presentations usually provide more than one view. The user can choose between timeline-based, graph-based, and structure-based editors. The same multimedia document is presented in different ways. Many of the tools do not provide an overview, be it for the whole video or for single elements. This may make it hard for ordinary end users to keep track of their elements in a large project. Some features from video annotation tools can be integrated in our authoring tool. An annotation overview on the player side, as seen in Overlay.TV, can be useful. It does not require any authoring interaction because it should be generated by the player in an automated way, if favored by the author.

2.3.2. Authoring Tools for Hypervideos

First authoring tools for hypervideos were designed and implemented in the 1990s, but most of the tools found in the literature are from the early 21st century. In addition to higher computing power, more sophisticated video formats and editing tools became available. They provided more abilities to create appealing presentations. Most of the tools found for the creation of hypervideos are described in scientific papers, only some web or commercial tools could be found. The editing of hypervideos needs more advanced authoring concepts than the editing of clickable, non-linear, and interactive videos. The editing process of hypervideos

[11] http://www.adobe.com/products/creativesuite/production.html (accessed January 2, 2013)
[12] http://www.adobe.com/products/director/ (accessed April 26, 2014)
[13] http://www.microsoft.com/silverlight/ (accessed April 26, 2014)
[14] http://www.microsoft.com/expression/ (accessed April 26, 2014)

is very similar to that of annotated interactive non-linear videos. Thus, hypervideo authoring tools are analyzed in more detail. The tools found vary widely in their characteristics and can be described in detail as follows:

HyperProp [SRMS00] "stresses the importance of document logical structuring and considers the use of compositions in order to represent context relations, synchronization relations, derivation relations and task relations in hypermedia systems. It discusses temporal and spatial synchronization among multimedia objects" [SRMS00]. No main medium is used, each type of medium like text, graphic, audio files, or video can be linked with each other using hyperlinks. Thereby, a non-linear structure is created. Neither alternative playback paths (jumps), nor an influence on the order of scenes before playback are given. The use of hotspots is not possible. The description language is NCL [Tel11]. The editor offers a structural view for the editing of the link structures, a timeline-based view for the definition of the temporal relationships, and a spatial view for the arrangement of the layout of the resulting presentation.

Advene [AP05; AP07; APS12] is a tool for active reading in videos. "One of the results of active reading applied to audiovisual material can be hypervideos, that we define as views on audiovisual documents associated with an annotation structure" [AP05]. One linear audio-visual document (video) is used as a main medium, annotations are rendered to different views. "An annotation type possesses a name and defines a content-type for its annotations, in the form of a MIME type (text/plain, text/xml, application/pdf, audio/wav, etc.)" [AP05]. Alternative playback paths (jumps) in the audio-visual document are definable by the annotation layer. Influence on the order of scenes can be given, but it depends on the annotations defined for the video. The definition of hotspots is not possible. A self-defined model and description format for the projects is used. GUI elements are a stream-based view, a view for note taking, a tree view, parallel time lines, a description area, and a video area.

HyperVideo Linking Generator (HVLG) [Hun97] is "a hypervideo system generator for automatic implementation of various hypervideo systems" [Hun97]. The main medium is video, several forms of annotations like image, sound, audio files, and video are possible. A non-linear structure can be defined using hyperlinks. Alternative playback paths (jumps) are implemented by specifying the frame numbers of the jump destination. Forks in the video are implemented by hotspots which are defined as clickable rectangles positioned in a fixed area for a defined frame range. Furthermore, hotspots are used to jump to annotations like video, audio files, images, and sound. An influence on the order of scenes cannot be applied. A self defined description model and structure ("Hyperlink data structure", "Generalized Hypervideo System Model (GVHS)" [Hun97]) are used. The GUI provides a video preview as well as a tabular view for links and a hotspot list.

Chang et al. [Cha+04] present an "object-based hypervideo authoring system. Video objects can be described by semantic annotation and multistory movies can be produced" [Cha+04]. The projects are based on one linear video which is enhanced with "multimedia descriptions" [Cha+04]. Thereby "additional data can be a text, a video clip, a URL link, or a still image" [Cha+04]. Hotspots are used as choice elements in the video to jump to other scenes which creates a non-linear link structure. Therefor annotated regions in a segment are chosen to be "branch points" (forks) [Cha+04]. No informa-

tion about the description language is given. The GUI provides a graph view, a video preview, and an overview for defined video parts.

HyPE and Jeherazade [HH06] are combined to implement "narrative intelligence in hypervideo" [HH06]. Jumps in linear videos are triggered by hotspots. Furthermore, hotspots are used to trigger the display of additional information like video, audio files, text, and images. Neither a non-linear structure nor an influence on the order of scenes can be implemented. A self defined XML file is used to describe the structure of the hypervideo. A video view and a list with hotspots (polygons) are part of the GUI.

Hsu et al. [Hsu+05] describe a tool for "hyper-interactive video browsing by a remote controller and hand gestures" [Hsu+05]. Video scenes are arranged in a graph structure. This non-linear structure can be navigated with "hyperlink[s] in a specified temporal-spatial domain" [Hsu+05]. Additional information are "text descriptions, existing image files, web page files or URLs on the Internet" [Hsu+05]. Neither alternative playback paths (jumps) nor an influence on the order of scenes is given. Furthermore, it is not possible to create hotspots in a scene. The tool contains a video preview, an annotation area, and a graph view.

Hyper-Hitchcock [SGW03b; SGW03a; SGW05; SGW08] is an authoring tool for the creation of detail-on-demand video. "Detail-on-demand video is a form of hypervideo that supports one hyperlink at a time for navigating between video sequences" [SGW08]. The main medium is video, additional information is also provided as video. A non-linear structure is defined by several types of links (detail links, prerequisite links, related information links, alternate view links, action choice links) [SGW08]. Alternative playback paths (jumps) are defined by links and user behavior. Choice elements for navigation between the video scenes are the key frames of the linked videos. It is not possible to define hotspots for navigation. The format of the internal link structure is not described in any of the papers. A timeline, a clip selection panel, a tree view, and a workspace area are part of the GUI.

Zhou, Gedeon, and Jin [ZGJ05] present a system for "automatic generation of additional information and the integration of the additional information to its corresponding selectable video object" [ZGJ05]. The outcome is a limited form of hypervideo, called detail-on-demand video. The main medium is a video which is annotated with video frame images and HTML files. Latter ones might contain links to further information. No choice elements or hotspots can be created with the described tool. Structure and relations between the elements are described in MPEG-7 [ISO09] which is converted to SMIL [W3C12]. The GUI is implemented as a converter view with two tree structures.

Finke and Balfanz [FB04] (partially based on [Bal+01]) describe "basic functional building blocks" of a "generic hypervideo concept" [FB04]. They furthermore derive a reference architecture from this concept which is then implemented as a prototypical example. The main media are videos which can be linked with other videos. Rectangled hotspots are defined which show additional information after user interaction. The prototypical implementation does not allow the editing of annotations, these can only be added as already edited files. The system is web-based. The description format for the metadata of the annotations as well as the format to describe the links between single video nodes is not described in any paper. The GUI depicted in [Bal+01] shows a tree view with keyframes and an editor to place rectangled clickable areas on a frame.

Klynt [Hon13] is a web platform for visual storytellers. The desktop editor has a visual storyboard to create a scene graph consisting of video scenes or multimedia pages. Multiple media formats can be added to the videos which are then played in an HTML5 player. Clickable buttons are added to scenes which show additional information mainly consisting of text and images or they load another scene. The integration of Facebook, Twitter, and Google maps is possible. The tool provides an WYSIWYG editor for links, a scene graph editor, annotation editors, and a timeline.

LinkedTV respectively VideoHypE [RGT13; BBO13] is a tool for "supervised automatic video hyperlinking". A video can be selected for which then are shots defined. These are arranged in an overview. Chapters can be defined from the shots. Furthermore, it is possible to select, name, and categorize entities. Therefor, hyperlinks can be specified which link to websites. The tool provides a timeline view. VideoHypE is part of the implementations of the LinkedTV project, for further information see [Lin13].

Critical reflection: The tools described in this subsection are capable of producing annotated interactive non-linear videos to a certain extent. They do not provide all functions needed to create an entire annotated interactive non-linear video like editors for a table of contents, tools to create navigation elements like button panels or quizzes, and forms to edit keywords for scenes and annotations. Furthermore, most of the tools were implemented to show new annotation principles or to combine editing principles, and therefore usability was rarely taken into account. In detail, HVLG, VideoHypE, and HyPE provide GUIs which give no overview of the structure of the whole video, which makes the authoring of larger projects difficult and reduces the usability of these tools. Klynt in contrast provides a GUI similar to those known from Adobe products like Adobe Premiere. It furthermore provides a scene graph view, where scenes can be linked with hyperlinks. The system described by Zhou, Gedeon, and Jin converts MPEG-7 files to SMIL files. It is no authoring tool in the traditional sense of the term, because it is not possible to compose different media files to an overall presentation. The GUI does not provide functions to link different media files in the GUI manually. Hyperprop has a different underlying structure compared to annotated interactive non-linear videos which is not based on video as a main medium. All types of media (like images or sound) can be used as a main medium in this tool. Furthermore, it provides only basic forms of navigation and does not allow the creation of a table of contents or a keyword list. Hsu et al. present a tool with focus on new input devices (remote controller, hand gestures) which allow navigation between video scenes arranged as a graph. This tool does not provide any form of advanced navigation. Advene allows to create hypervideos based on active reading (annotating) and generated rules. Thereby, different new authoring paradigms are proposed. VideoHypE is a similar tool, which mainly focuses on the annotation and the hyperlinking of video segments. The tool described by Chang et al. uses one linear video in which hotspots are defined to jump to points on the timeline. Accordingly, if different videos are needed, they have to be merged to one single video before editing, which decreases usability and requires that the user merges the parts in another tool. Hyper-Hitchcock allows the creation of non-linear structures, but the only type of medium that can be used is video. This contradicts the intention of creating rich media presentations consisting of different types of media. The system described by Finke and Balfanz allows the creation of link structures and hotspots, but provides only basic editing functions for annotations. This requires the usage of other tools even for smaller editing tasks, which is not very user-friendly. All four authoring paradigms from [BH05] can be found in tools and systems for hypervideos. The range of functions varies between the

tools. Accordingly, no general structure of an editor can be stated for hypervideo authoring tools. (Parts of this subsection were taken and adapted from our previous work [Mei+12b].)

2.3.3. Summary

In this section, we analyzed authoring tools from different areas. Tools for hypervideos are capable of producing annotated interactive non-linear videos to a certain extent, but they do not provide all functions like editors for a table of contents, tools to create navigation elements like button panels or quizzes, and forms to edit keywords for scenes and annotations. Usability was rarely taken into account during the implementation phase in related work. All four authoring paradigms from [BH05] can be found in tools and systems for hypervideos. In addition to that, each tool provides a different range of functions. Because of that, no general structure of an editor can be derived from the hypervideo authoring tools. Tools from the other areas provide some of the needed editors and functions and can be used to analyze the user interfaces. Used description languages and standards can be found in Table 2.1. Many tools use self-defined languages (often XML format) or the authors do not describe the description language in their work. SMIL, a standard for multimedia presentations is used by some tools from this area, but not by all of them. This is partially because first authoring tools were implemented before the standardization of SMIL. (Parts of this subsection were taken and adapted from our previous work [Mei+12b].)

	Clickable video	Non-linear video	Interactive video	Hypervideo	Multimedia presentation
SMIL				Zhou et al. [ZGJ05]	SMIL Builder [BB11], GRiNS [Bul+98], Java-Assisted SMIL (JAS) [DTL06], Jokela et al. [JLK08], Sung and Lee [SL05], SMILAuthor [YY03]/ SMILAuthor2 [YCW04a; YCW04b; YCW08]
NCL			Composer [LGaM-DRCGS08]	HyperProp [SRMS00]	
HTML + JavaScript			Popcorn Maker [Moz13a]	Klynt [Hon13]	Matchware Mediator 9 [Mat12]
MPEG-7			SeViAnno [Cao+10]	Zhou et al. [ZGJ05]	
self-defined/ other		Riva Producer Enterprise [mem13], XIMPEL [Bhi+10]	LazyMedia [HL06], HyStream System [Bea+02], Chen et al. [Che+09a], Räck et al. [RSA10], ConnectME [NBB13]	Advene [AP05; AP07; APS12], HyperVideo Linking Generator (HVLG) [Hun97], HyPE and Jeherazade [HH06], VideoHypE [RGT13; BBO13]	CMIFed [Ros+93], Shih et al. [Shi98; Shi+98b; Shi+98a], the Synchronization Editor [BHL92], Cutts et al. [Cut+09], Deng and Shih [Den+02b], Harmony [Fuj+91], Gaggi and Celentano [GC02], MEMORY [KHM08], IMMPS [SD97], Villard [Vil01], MPRES Author [WRR97], Madeus [Jou+98], SIMPLE [Mur+06], Eventor [Eun+94], MediaTouch [Ech+98]
not described	Klickable [Kli13], Overlay.TV [Ove10], VideoClix [Vid12], Viddix Beta [VID10]	YouTube Video Annotations [You13]	wireWAX [Wir12], HyLive [HKH08], Chang et al. [CHS07; CHC08], 5minMedia VIDEO EVERYWHERE [5mi14], Zodiac [Chi+00], ADIVI Production Kit [Inn11], Quicktvpro [Bel12]	Chang et al. [Cha+04], Hsu et al. [Hsu+05], Hyper-Hitchcock [SGW03b; SGW03a; SGW05; SGW08], Finke and Balfanz [FB04]	Delaunay MM [CL97], TYRO [Mac91], LECTURNITY 4 [imc10], NextSlidePlease [Spi+12]

Table 2.1.: Overview over description languages and standards used in the authoring tools described in related work.

2.4. Evaluation of Existing Players

While standard controls like play, pause, stop, fast-forward, fast rewind, volume control, or a timeline are sufficient for traditional linear videos, players for videos with extended functions require more advanced concepts for display and interaction. Jain claims that "in the field of entertainment and training, where interactive video is expected to be useful, much more friendly interface is desired [sic]" [JW95]. Contrary to authoring tools which are used by a small number of authors, players are used by a much wider range of viewers with different skill levels. This requires an intuitive arrangement and labeling of buttons as well as interactive elements like hotspots depending on the type of video. Many authoring tools offer some kind of own player implementation, which is described in this section. These optimize the output of the content created in the authoring tool and provide functions suited for the desired use cases. The categorization of the players described in this section is based on the categorization of the corresponding authoring tools and/or multimedia models and resulting description languages (which are not described with the players if already qualified with the authoring tool). Players for hypervideos are described in more detail like the according authoring tools. All players mentioned in this section are listed in tabular form in Appendix D on page 239. Players described in scientific work were not tested for the use with current operating systems.

2.4.1. Other Players

Player for **clickable videos**, namely those provided by Klickable [Kli13], Overlay.TV [Ove10], VideoClix [Vid12], ConciseClick [Cle12], and Viddix Beta [VID10] are without exception web players. All of them play linear videos and offer no controls to select other paths through the video accordingly. Hotspots show additional information, mainly text, images, and links which are positioned as overlays over the video (Klickable, VideoClix, ConciseClick, Viddix Beta) or in a two part video view (Viddix Beta). The clickable area of the hotspots is represented as an image of an object in the video in Overlay.TV, as a shape of an object using VideoClix, as rectangles in Viddix Beta and Klickable, or not at all in ConciseClick. Player controls are play/pause, timeline, and volume control in all of these tools. In addition, Overlay.TV provides an info and a share button. The VideoClix player is equipped with buttons and menus for full-screen, a list of objects, share, settings, and recommended videos. A list of objects from the video is displayed in the ConciseClick player, too. The Viddix Beta player only provides a full-screen in addition.

All authoring tools for **interactive videos** (except Zodiac [Chi+00]), Composer [LGaMDRCGS08], wireWAX [Wir12], Popcorn Maker [Moz13a], HyStream System [Bea+02], 5minMedia VIDEO EVERYHWERE [5mi14], ADIVI [Inn11], SeViAnno [Cao+10], HyLive [HKH08], Quicktvpro [Bel12], LazyMedia [HL06], as well as the tools described by Chang et al. [CHS07; CHC08], Chen et al. [Che+09a], and Räck, Seeliger, and Arbanowski [RSA10] are implemented in combination with players. The players are either web players like in wireWAX, Quicktvpro, Räck, Seeliger, and Arbanowski, Popcorn Maker, 5minMedia VIDEO EVERYHWERE, LazyMedia, and ADIVI or stand alone players as in Chang et al., Chen et al., and HyLive. Browser plugins for YouTube[15] and Brightcove[16] are provided by wireWAX and

[15] http://www.youtube.com/ (accessed April 26, 2014)
[16] http://www.brightcove.com/en/ (accessed April 26, 2014)

Quicktvpro. Furthermore, a plug-in for Viddler[17] is available from wireWAX. Quicktvpro offers player plug-ins for Kaltura[18], OOYALA[19], Longtail[20], and Vzaar[21] in addition. Many players are equipped with a set of standard buttons (play, pause, etc.), for example the players of Chang et al., Räck, Seeliger, and Arbanowski, Popcorn Maker, HyStream System, SeViAnno, and Quicktvpro. More extended button sets and menus are provided by the 5minMedia VIDEO EVERYHWERE and the Quicktvpro player. Jumps in the video are possible by clicking on markers on a timeline or on clickable annotations in the players of wireWAX, SeViAnno, Quicktvpro, LazyMedia, Chang et al., and 5minMedia VIDEO EVERYHWERE. Clicking on hotspots triggers additional information in wireWAX, Räck, Seeliger, and Arbanowski, ADIVI, and HyLive. Hotspots may also link to other websites as in Popcorn Maker, 5minMedia VIDEO EVERYHWERE, and Quicktvpro. Chang et al. implemented interactive buttons and images. Chen et al. use interactive games to allow the user to access the next scene. Annotations can be displayed in three ways of spatial arrangement: Overlays over the main menu are used by wireWAX, Chang et al., Räck, Seeliger, and Arbanowski, 5minMedia VIDEO EVERYHWERE, and ADIVI. Separate areas for the main video and the annotations are used by 5minMedia VIDEO EVERYHWERE, LazyMedia, HyStream System, and SeViAnno. Video area and annotations are freely positioned in the player of the Popcorn Maker. A different form of interaction is provided by the EmoPlayer as described by Chen et al. in [Che+08]. This player allows the selection of a character in the video and watch affective annotations indicating its emotions. A mixture of player and collaborative authoring tool is described by Franzoni, Ceballos, and Rubio in [FCR13]. They use a linear video in a web platform to which users can add new annotations. Furthermore, the video can be randomly accessed by questions or search targets which enables jumps in the video.

Players for **non-linear videos** like XIMPEL [Bhi+10], the Riva player [mem13], and the YouTube player [You13] are all implemented as web players. The Riva player is available as a stand-alone player in addition. The graph structure is realized as video scenes in XIMPEL and the Riva player. YouTube Video Annotations uses several linear YouTube videos which are linked by invoking a website with the embedded video to build the graph structure. An overall impression of one video played in one player is destroyed thereby. Choice elements are implemented as hotspots which invoke/link to other scenes. Standard controls provided by XIMPEL and the Riva player are pause/play, fast-forward, and rewind. XIMPEL also provides a full-screen mode, the Riva Player a timeline and a volume control. The YouTube player offers buttons for play/pause, to hide annotations, to change the video quality, for watching later, to change the player size, as well as a volume control and a timeline. Annotations may be available in the form of text (with some kind of framing) and images which are displayed as overlays over the video area.

Multimedia presentations can be viewed with the player implementations of GRiNS [Bul+98], Blakowski, Hübel, and Langrehr [BHL92], CAI application (Eventor) [Eun+94], Gaggi and Celentano [GC02], Jokela, Lehikoinen, and Korhonen [JLK08], Cutts et al. [Cut+09], Deng and Shih [Den+02b], MEMORY [KHM08], TYRO [Mac91], IMMPS [SD97], MPRES Viewer [WRR97], SIMPLE [Mur+06], Madeus [Jou+98], LECTURNITY 4 Player [imc10], NextSlidePlease [Spi+12], Chrooma+ [Oeh+13], and CMIFed [Ros+93]. No player is offered for authoring tools like SMIL Builder [BB11],

[17] http://www.viddler.com/ (accessed April 26, 2014)
[18] http://corp.kaltura.com/ (accessed April 26, 2014)
[19] http://www.ooyala.com/ (accessed April 26, 2014)
[20] http://www.longtailvideo.com/ (accessed April 26, 2014)
[21] http://vzaar.com/ (accessed April 26, 2014)

SMILAuthor [YY03], Shih et al. [Shi98; Shi+98b; Shi+98a], The Synchronization Editor [BHL92], Delaunay MM [CL97], Java-Assisted SMIL (JAS) [DTL06], Harmony [Fuj+91], Sung and Lee [SL05], Villard [Vil01], and the Matchware Mediator 9 [Mat12]. The outcome of these authoring tools is either presented with SMIL players like AMBULANT SMIL (2.0/2.1/3.0) player [CWI10; Bul+04], RealPlayer 16 (SMIL 2.1) [Rea12], or with standard web browsers. The players of GRiNS, TYRO, NextSlidePlease, SIMPLE, CMIFed, and Jokela, Lehikoinen, and Korhonen are implemented as a preview in the authoring tool or they are a combination of player and authoring tool. No stand-alone player is available. Players in Blakowski, Hübel, and Langrehr, Cutts et al., Eun et al., Madeus, the LECTURNITY 4 Player, the AMBULANT SMIL player, and the RealPlayer 16 are implemented as stand-alone desktop players. Furthermore, a browser plug-in is available for the AMBULANT SMIL player. Web players or presenters are described for Deng and Shih, MEMORY, and MPRES Viewer. No player interface was implemented in the work of Gaggi and Celentano. They propose an execution simulator with special controls for the simulator. Media are represented by placeholders. Dealing with multimedia presentations, more interaction is possible with the player itself than with contents or hotspots of the presentation. Only few players provide navigational and interactive features like a table of contents (Cutts et al.), a graph structure which illustrates the current position (NextSlidePlease), a search function, markers on a timeline (Cutts et al., LECTURNITY 4 Player), hyperlinks (Gaggi and Celentano, LECTURNITY 4 Player, CMIFed, NextSlidePlease), a list of jump destinations (MEMORY, LECTURNITY 4 Player, NextSlidePlease), or buttons/hotspots (IMMPS, LECTURNITY 4 Player). In players that interpret SMIL files like the AMBULANT SMIL player or RealPlayer 16, interaction and navigation are strongly dependent from the functionality provided by the authoring tool. Depending on the underlying synchronization model, player controls are either for a whole presentation or for single types of media. Standard controls like play/pause, fast-forward, rewind, stop, and restart for a whole presentation are described by Blakowski, Hübel, and Langrehr, Cutts et al., Jokela, Lehikoinen, and Korhonen, for the LECTURNITY 4 Player, in CMIFed, for the AMBULANT SMIL player, and for the RealPlayer 16. Controls for single media or channels are implemented in the MEMORY and the SIMPLE player. Scenario dependent controls are described for IMMPS. Annotations are presented in a fixed arrangement around some kind of main medium in Cutts et al., Jokela, Lehikoinen, and Korhonen, in MEMORY, or the LECTURNITY 4 Player. They are presented as overlays over the video or in side areas in the Chrooma+ approach. All media files are arranged independently in TYRO, IMMPS, SIMPLE, CMIFed, AMBULANT SMIL player, and RealPlayer 16.

Critical reflection: Players for non-linear videos use player technologies like the Adobe Flash Player[22], the Shockwave Player[23], the Quicktime Player[24], the Windows Media Player[25], or Microsoft Silverlight[26]. Hotspots are used to invoke other scenes. These players have standard controls, timelines, and buttons for full screen, to hide annotations/links, to change the video quality, and for bookmarking. Annotations/links are displayed as overlays over the main video. Interactive video players provide hotspots, interactive images, and interactive games. Some players offer an extended set of buttons which are not always intelligible to users [Mei+12c]. Three different ways to arrange annotations exist: Overlays over the main

[22] `http://www.adobe.com/products/flashplayer.html` (accessed April 26, 2014)
[23] `http://www.adobe.com/products/shockwaveplayer.html` (accessed April 26, 2014)
[24] `http://www.apple.com/quicktime/` (accessed April 26, 2014)
[25] `http://windows.microsoft.com/en-us/windows/windows-media-player` (accessed April 26, 2014)
[26] `http://www.microsoft.com/silverlight/` (accessed April 26, 2014)

video area, two separate fixed areas for video and annotation, and a number of freely positionable areas for video and annotations are possible. Players for clickable videos provide hotspots to display additional information which is then positioned as overlay over the video or in a two part view. Besides standard player controls, buttons and menus for full screen, a list of objects, and others exist. Multimedia presentations mainly have standard controls for navigation in the whole presentation. Some players provide navigation for single media or media channels and hyperlinks to other parts of the presentation.

The player for annotated interactive non-linear video needs clickable hotspots as well. Furthermore, buttons for full screen, to hide/show annotations, and for bookmarking seem useful. The implementation as a web player allows a large number of users to watch videos without the need to install additional software on their PCs or mobile devices.

2.4.2. Players for Hypervideos

Navigation and player controls vary widely in player implementations for hypervideos. Some new controls and navigation elements are shown in Joscha Jägers prototypical implementation of the "Open Hypervideo Player" which was implemented in HTML5, JavaScript, CSS, and SVG [Jae12]. Most players (except those described otherwise below) were implemented as standalone players for desktops. All authoring tools described in Section 2.3 except the HVLG [Hun97] and the tool described by Zhou, Gedeon, and Jin [ZGJ05] provide their own player implementations:

HyperVideo Linking Generator (HVLG) [Hun97] offers a player in which "the viewer can trigger a hyperlink and jump from frame to frame" [Hun97]. A non-linear structure is defined by hyperlinks and jumps to frame numbers in or between video scenes. The user clicks on rectangled hotspots which have a fixed position for a defined frame range. The GUI of the player offers standard controls. Images and videos are displayed in the main area of the player, sound can be played as well.

HyperProp [SRMS00] is described as follows: "The spatio-temporal formatter, or simply formatter, is responsible for controlling the document presentation based on its specification and on the platform (or environment) description" [SRMS00]. The "document" contains media files in a certain structure which is defined in NCM/NCL [Tel11]. The formatter is implemented in Java[27], but no detailed description of the user interface is given.

Hyper-Hitchcock player [SGW03b; SGW03a; SGW05; SGW08]: This player is realized as a stand-alone player for detail-on-demand videos. "The Hyper-Hitchcock player was iteratively developed over several user studies. These studies emphasized efficiency of access to information in the hypervideo. The resulting design included keyframes and link labels to help viewers rapidly navigate and orient themselves" [SGW08]. A non-linear structure and alternative playback paths are defined by several types of links which are represented by the keyframes of linked videos. "All keyframes are clickable, thus enabling the user to return several link levels at once" [SGW08]. This form of video offers no clickable hotspots in the video. Controls of the player GUI are buttons for play, stop, and navigation as well as a timeline with a keyframe preview. All videos are displayed in the main video area.

[27] `http://www.java.com/en/about/` (accessed April 26, 2014)

Chang et al. [Cha+04] present a video player which is "developed for the video viewer to view the annotated film efficiently" [Cha+04]. One linear video is used, the non-linear structure is based on annotations. Alternative playback paths (jumps) can be selected by clicking on a 'branch point' (annotated region in a segment). These regions defined as hotspots are used for jumps in the video (to other scenes). Implemented player controls cannot be determined from this work. Additional information are "multimedia descriptions" [Cha+04], or more precisely "a text, a video clip, a URL link, or a still image" [Cha+04]. The snapshot in this work shows the additional information in a two-part GUI in the right area.

Finke and Balfanz [FB04] present a web video player which consists of three areas for display. Besides a video area, a navigation view and an information and communication view are available. Video scenes in a linear order are used. Hotspots are used to invoke information about objects in the video. These are then displayed as HTML pages in the information view. Player controls are play/pause, jump forward/backward between scenes, and a timeline. The navigation view provides links to multimedia annotations.

Advene [AP05] provides two implementations/views for playback. A static view can be described as a "definition of a hypertext document, whose temporality is imposed by the user visualising [sic] it" [AP05]. In a dynamic view "the temporality of the resulting document is mostly imposed by the audiovisual document. Of course, they also offer some kind of interaction opportunities and the user normally always has the possibility to interrupt playing, [...] but we can imagine kiosk-like approaches where all video controls are deactivated" [AP05]. Thereby, one audio-visual document can be navigated via timeline or URLs. Neither choice elements which have influence on the video structure nor hotspots provide interactivity to the viewer. The GUI of the player offers standard controls, hyperlinks, an URL stack, navigation links, and a position indicator for navigation in the video. Annotations are shown around the video and as overlay over video. They are mainly text-based.

Hsu et al. [Hsu+05] follow a different idea. They describe "hyper-interactive video browsing by a remote controller and hand gestures" [Hsu+05]. Non-linear video with a graph structure offers hyperlinks "in a specified temporal-spatial domain" [Hsu+05]. These links are navigated by gesture controls. Additional information like text descriptions, existing image files, web page files, or URLs on the Internet can be displayed with the video.

HyPE stand alone player [HH06]: "The hypervideo player loads and starts the basic video and the meta data information" [HH06]. Based on a linear video, non-linearity is implemented by jumps. These are triggered by hotspots in the video. Furthermore, hotspots are used to display additional information. The GUI is implemented as a two-part window with a video or audio player on the left side, and a text or an image viewer on the right side. No controls are offered by the GUI.

Klynt [Hon13]: The Klynt player is implemented as a Web player in HTML5. It provides different customized buttons as overlays on the video for navigation between video scenes and for the display of annotations. These may contain more buttons, images, text, videos or other web based contents. Furthermore, it is possible to add a Google maps menu consisting of a map with markers which are then linked to video scenes. Other navigational elements are presentation-like screens with buttons to other screens

or to video segments. These elements allow the creation of hypervideos with focus on different media types.

LinkedTV [OL13]: The player from the LinkedTV project is designed as a second screen application for desktops, smart phones and tablets. The first screen is used to play the video while the second screen can be used to control the main screen. Furthermore, it is possible to navigate to another chapter in the presentation. The second screen has an interface which shows "detected entities of the video grouped by persons, objects and locations" [OL13]. Different external control interfaces are available or under development.

Critical reflection: The players described in this section show differences in presenting the additional information and in provided controls. HVLG, the Hyper-Hitchcock player, Klynt, and the player presented by Hsu et al. have one single area, where the main video and additional information are shown. The latter are either shown as smaller overlays or they replace the video. The player described by Chang et al. and the HyPE stand alone player are implemented as two-part windows. The video is played in the first and annotations are presented in the second area. The annotations are grouped in different areas around the main video or may be displayed as overlays in the video area in Advene. LinkedTV provides a second screen where annotations are shown according to the contents of the first/main screen. Standard controls like play and pause are implemented in Advene, Klynt, HVLG, and the Hyper-Hitchcock player. Advene provides different links and a position indicator in addition. Furthermore, the Hyper-Hitchcock player offers a timeline with keyframe preview. Rectangled hyperlink areas are implemented in HVLG and Klynt. The player described by Hsu et al. has no clickable controls, the interaction is done by gestures and a remote controller. No controls are implemented in the HyPE stand alone player, navigation is only possible with elements embedded in the video. Menu-like structures can be displayed in the Klynt player. Different elements and functions described in this section are needed in our player for annotated interactive non-linear videos, too. Important features are clickable hotspots, basic player controls, and menu-like structures.

2.4.3. Summary

Players from the varying areas differ in presenting the additional information and in provided controls. Additional information is shown in annotation areas beside the video or as overlay on the video area. All players provide at least some of the standard VCR-controls, others have an extended button set for different new functions. More advanced players have extended timelines, hotspots, or additional navigational elements. Players are implemented as web players, stand-alone players, or browser plug-ins.

Players for non-linear videos are implemented as web players or stand-alone players (light blue fonts in Figure 2.4). Interactive video players are mainly implemented as web players or as browser plug-ins (dark blue fonts in Figure 2.4). Players for clickable videos are without exception web players (green fonts in Figure 2.4). Multimedia presentations (purple fonts in Figure 2.4) are displayed as previews in authoring tools, in a combination of player and authoring tool, in stand-alone desktop players, as browser plug-ins, or as web players.

Our player for annotated interactive non-linear video needs clickable hotspots as well. Furthermore, buttons for full-screen, to hide/show annotations, and for bookmarking seem use-

ful. The implementation as a web player allows a large number of users to watch videos without the need to install additional software on their PCs or mobile devices. Hints on the implementation of a table of contents, a keyword search, or advanced navigational structures could be found rarely.

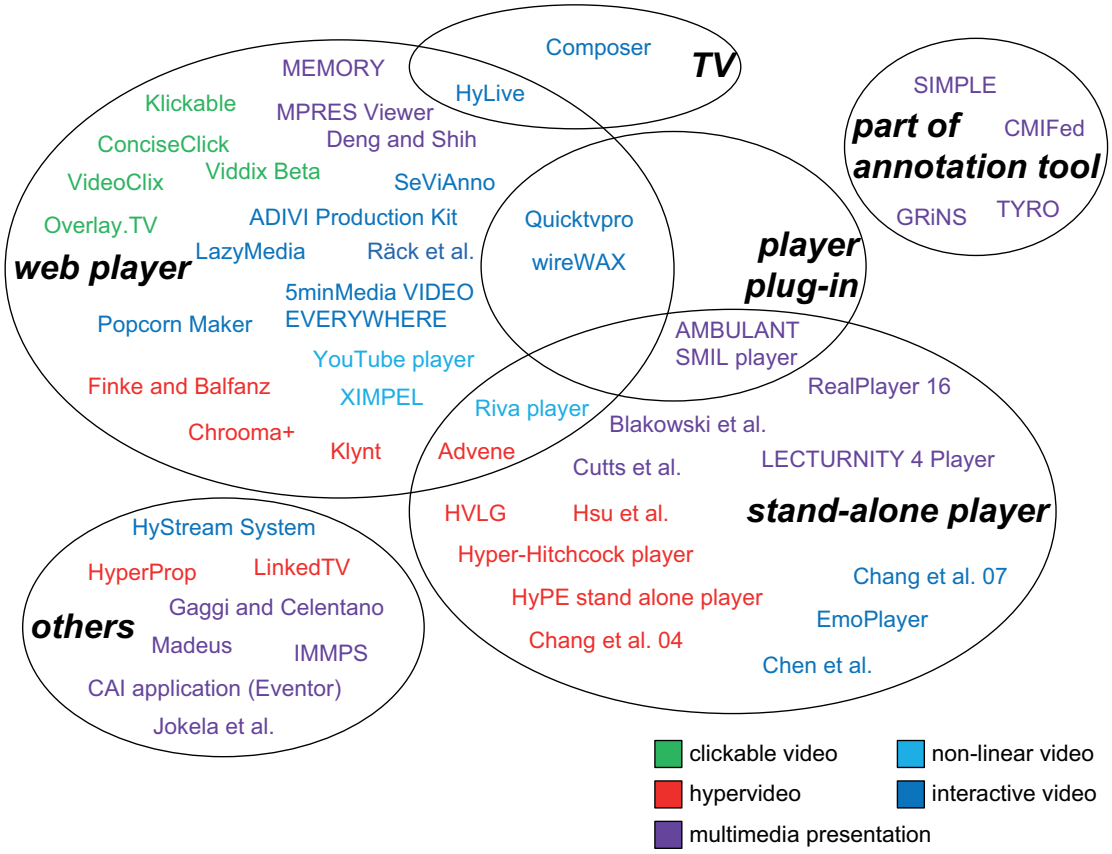


Figure 2.4.: Players categorized by their type of playback device/software and their classification used in this work.

3. SIVA Suite

Several functions and features of an annotated interactive non-linear video software suite are needed for the scenarios described in Section 1.1. We implemented the SIVA Suite (**S**imple **I**nteractive **V**ideo **A**uthoring Suite). The two major goals in the development process were, besides the coverage of all needed functions, a good expandability, and a high usability for non-professionals. The SIVA Suite was designed and implemented at the Passau University in four different projects which partially built upon each other[1]. The goal of the first project was the implementation of the basic functions in an authoring tool and a player, as well as to define a structure for an XML file. In the second project, collaborative elements were added to a web player. Furthermore, mobile players for smart phones were designed and put into practice. The third project focused on instructional videos and a web platform for the administration of the single video projects including a rights management. The goal of these projects was knowledge transfer to small and medium-sized enterprises (SMEs) which we accomplished with our easy to use software and its outcome on the one hand and with workshops about the software and its functions on the other hand. The last project, which will be finished in 2016, validates the software in different application contexts. Media design, effectiveness, usability, and legal aspects are taken into account thereby.

The SIVA Suite consists of three stand-alone components: A metadata format (XML file) for the definition of sequence control, interaction, and navigation, an authoring tool called SIVA Producer, as well as the playback component, the SIVA Player. The interaction of the single components can be described from two points of view, the data exchange and the data flow:

- **Data exchange:** The three components exchange data with each other in the following way (see also Figure 3.1): an author creates an annotated interactive non-linear video in the SIVA Producer and exports it into an XML file with its inherent media files. The SIVA Player reads the XML file and displays all media files as well as elements defined by the XML file according to its instructions.
- **Data flow:** The authoring tool is used to create the annotated interactive non-linear videos (Figure 3.2, (1)) which is then exported as a video project and uploaded to a web server (Figure 3.2, (2)). A client loads a website and starts a video in the player (Figure 3.2, (3)). The player then downloads the control file (Figure 3.2, (4)), processes it, and starts downloading necessary media files (Figure 3.2, (5a), (5b)). When a scene

[1]The development of the SIVA Suite was partially funded by the European Social Fonds and the Bayrisches Staatsministerium für Wissenschaft, Forschung und Kunst (Bavarian State Ministry for Sciences, Research and the Arts) under project names “Interaktives Video Editierungstool zum netzwerkbasierten Wissenstransfer (iviPro)” (“Interactive video editing tool for network-based knowledge transfer (iVi-Pro)”) and “iVi-Pro 2.0 - Interaktives Video im Zeitalter von Mobilität und Kollaboration” (“iVi-Pro 2.0 - Interactive video in the age of mobility and collaboration”) as well as “MobileTechTeach - mobile multimediale Hilfesysteme für technische Anwendungen in KMU” (“MobileTechTeach - Mobile multimedia help systems for technical applications in SME”). Furthermore, it was funded by the Bundesministerium für Bildung und Forschung (German Federal Ministry of Education and Research) (BMBF) under project number 03V0633.

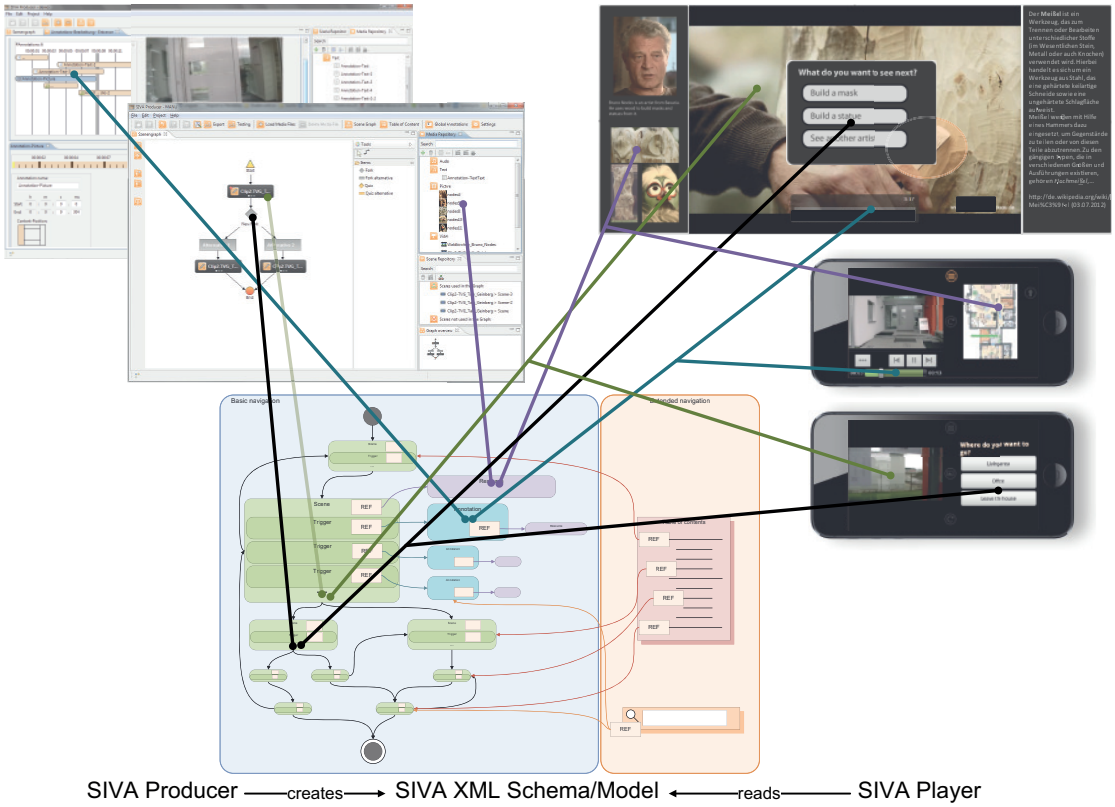


Figure 3.1.: Data exchange of the components of the SIVA Suite: the SIVA Producer creates an XML file, the SIVA Player reads the XML file and interprets its instructions.

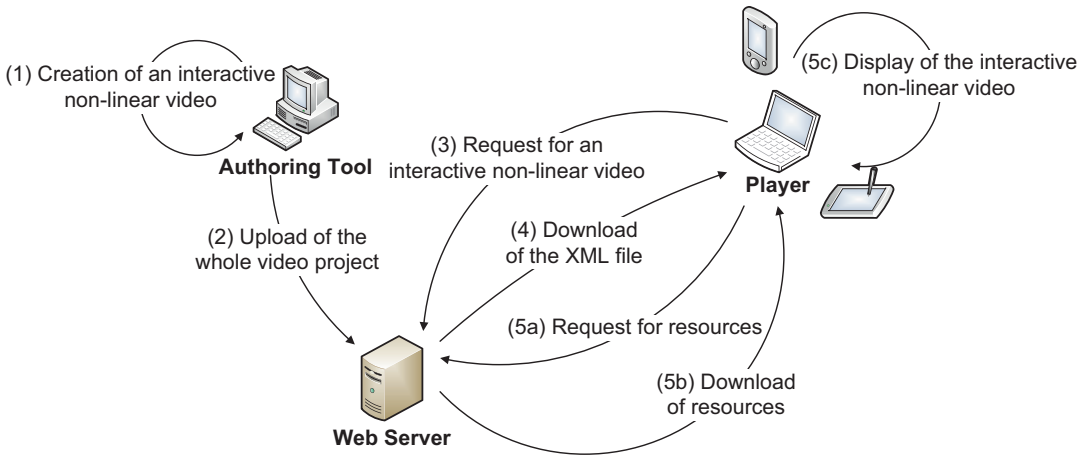


Figure 3.2.: Data flow in the SIVA Suite: the SIVA Producer exports a video project which is uploaded on a web server; the player requests a video, downloads first the XML file and then all other files needed for display.

is playable, it is displayed to the user (Figure 3.2, (5c)). At the end of a scene or when additional media are needed for display, steps (5a)-(5c) are repeated.

We split our functions and elements into three categories. They are needed to put the use cases described in Section 1.1 to practice. We also show how we implemented them in each component. **Interactivity** is achieved by clicks on objects in the video or by clicks on player elements. **Non-linearity** is realized by navigation in the structure of the scene graph with button panels or quizzes[2]. Another way of viewing the video non-linearly is by navigating through it by a table of contents or using a keyword search. **Annotations** may be rich text, images, audio files, or videos which are displayed and hidden at defined points in time. The definition of keywords is also considered as additional information in form of metadata. Both, non-linearity and additional information require some kind of interactivity. Some functions could be assigned to more than one of these categories. Furthermore, they are not implemented in all three components as shown in Table 3.1.

	XML	Producer	Player
Interactivity			
clickable hotspot marking an object	✓	✓	✓
clickable markers on timeline	–	–	✓
jump to other point in time in the video	–	–	✓
search a keyword	–	–	✓
VCR actions	–	–	✓
Non-linearity			
scene graph with	✓	✓	–
selection panels	✓	✓	✓
quizzes	✓	✓	✓
table of contents	✓	✓	✓
search	–	–	✓
Annotations			
index/keywords	✓	✓	–
display/hide information	✓	✓	✓
types of media			
rich text	✓	✓	✓
images	✓	✓	✓
audio files	✓	✓	✓
videos	✓	✓	✓

Table 3.1.: Implementation of interactivity, non-linearity, and additional information in the three components of the SIVA Suite.

As Fox already stated in 1989, "Interactive systems will generally be successful only if the human-computer interface is carefully developed according to principles of graphic design and cognitive psychology, and is thoroughly tested" [Fox89]. We developed the SIVA Producer and the SIVA Players in their web and mobile versions in an iterative process of development activities as well as functional and usability tests. As we described the shortcomings of existing tools and metadata formats in the previous chapter, we are now going to describe our

[2]The quizzes are not described in detail in this work. For further readings see [MGK11].

approach on implementing an XML schema/model, an authoring tool, and a player. Thereby we try to answer the following research questions:

How can content and control information be modeled for playback?

How can the composition of interactivity, non-linearity, and additional information be comprehensibly managed in an authoring tool?

How are interactivity, non-linearity, and the display of additional information realized in desktop and mobile players?

3.1. SIVA XML Schema: Modeling of Annotated Interactive Non-linear Video Behavior

We followed the design principles of SMIL [W3C12], which are the maintenance of a declarative XML format, the separation of content and structure, and the support of a flexible architecture [BR08]. Major design goals were the easy expandability and the slim format which exactly fitted our requirements as well as existing and potential future scenarios without too many limitations. We decided to implement some logic into the player to avoid repetitive definitions in the XML file. However, our format shows several differences in its internal structure compared to SMIL. Our XML format is event-based and provides simple time synchronization. The temporal structure of the format is easy to maintain because of its high level of modularity. SMIL defines the temporal sequence of elements which are displayed with each other in nested structures. Adding a new element may require changes in different parts of the XML file. The SIVA XML format is based on the assumption that one main video is displayed at a time. All elements displayed with this main video are triggered by its timing information and linked by ID/IDREF attributes. Due to the locality of the elements, single XML files are easier to extend. Furthermore, ID/IDREF attributes are checked by constraints for their consistency (for example in case of collaborative editing functions in a player). We chose the XSD instead of the DTD for the definition of the structure of our files to ensure the internal correctness of references and data types. Using the XSD allows us to define constraints which ensure a consistent definition of keys and references.

3.1.1. Conceptual Model

The conceptual model of our XSD illustrated in Figure 3.3 shows two ways of navigation in the video. Both implement *non-linearity*. The first one called “basic navigation” is based on the structure of the video corresponding to its scene graph. The second one called “extended navigation” is independent of the basic navigation.

Each annotated interactive non-linear video contains a set of scenes. Figure 3.3 shows them in green color. Each scene has a reference to a video **resource** (purple color in Figure 3.3). The separation between structure and content enables us to exchange the content, for example to provide different video qualities. This may be necessary for example to implement multilingualism, to provide different levels of quality, or to make different video formats available for different end user devices (more precisely for different display sizes and different bandwidths). Scenes are linked to each other in a **scene graph** (black arrows in Figure 3.3) which

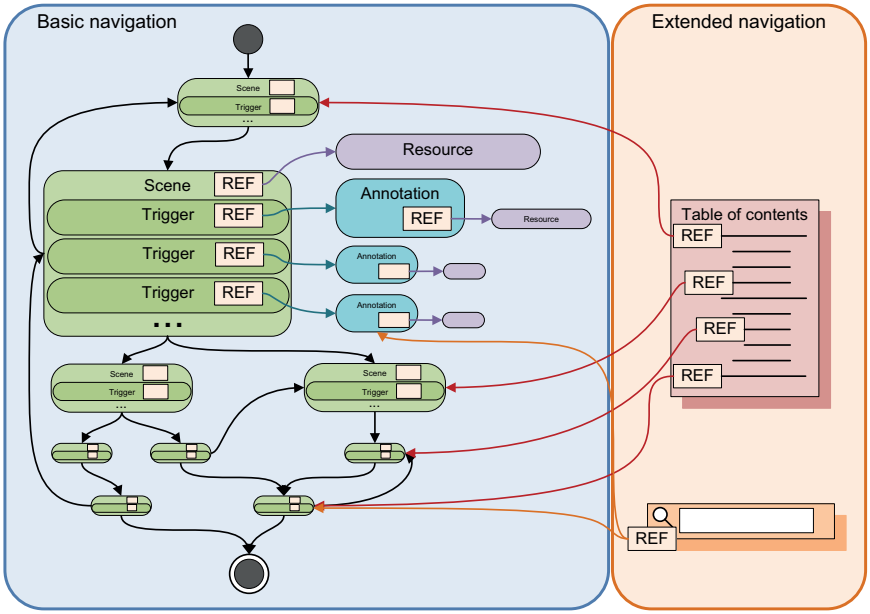


Figure 3.3.: Conceptual model of an annotated interactive non-linear video as defined in this work: basic navigation includes the links between scenes and annotations, extended navigation are a table of contents and a keyword search.

implements the concept of *non-linearity*. Furthermore, each scene graph has a **beginning**. The first scene is always connected in series with the start of the video. Each scene of the scene graph must have a path to the **end** and, if the edges are reversed, to the beginning of the scene graph. The start (source) and the end (sink) of the graph are illustrated in UML notation in Figure 3.3. **Annotations** (turquoise color in Figure 3.3) form a self-contained set, because the same annotation can be displayed more than once in the whole video as *additional information*. The annotation is linked to a scene by a trigger. All triggers have a defined start and an end time. Both of them belong to the same scene (except in case of a so called global annotation which is displayed during the whole playback of the video). An annotated interactive non-linear video contains two different types of annotation. The “standard annotation” is displayed without any user interaction. *Interactivity* is added by another type of annotation, called “interactive annotation”. It is defined in the same way as the standard annotation but has a clickable area, for example a rectangle marking, an object in the video, or a button on a button panel.

One optional element to add *non-linearity* is a **table of contents**. One entry consists of a link to a scene and a caption which is displayed to the user. The conceptual model (Figure 3.3) shows the table of contents in red color. It contains references to four different scenes of the video indicated by red arrows. The second optional *non-linear* element is a **keyword search**. It contains a search string and a set of one or more targets. A target can point to the start of a scene or to the start of the display of an annotation in a scene. The element for the keyword search is depicted in orange color in Figure 3.3. The inserted word would match for a scene and an annotation indicated by orange arrows.

3.1.2. XML Schema and XML File

The XML file compliant to our XSD can be divided into the following six parts: the **project information**, a **scene list**, **resources**, **actions**, a **table of contents**, and a **list of keywords** linked to annotations or scenes (see Figure 3.4). *Non-linearity* is mainly implemented in the scene list, the table of contents, and the list of keywords part. *Interactivity* is realized in the actions part. Definitions for *additional information* can be found in the project information, the resources, and the actions part. Due to the internal references (ID/IDREF), no clear assignment of the parts to one of the characteristics is possible. We describe the parts of the XML file as independent sections hereafter.

We chose to separate structure and contents in the XML format itself. The scene graph defines the primary structure of the video. Each scene contains triggers for actions which may show annotations or invoke interactive elements. The synchronization issues are limited to the scene, which makes it easy to schedule downloads and deal with intra scene user interaction. All internal and external resources are defined in a particular section which makes it easy to maintain the files in several languages. Additional sections for a table of contents and a keyword search enhance the modularity and with that maintainability and expandability. The XML Schema and its corresponding XML files were benchmarked for their performance by Janda [Jan10; Mei+10b].

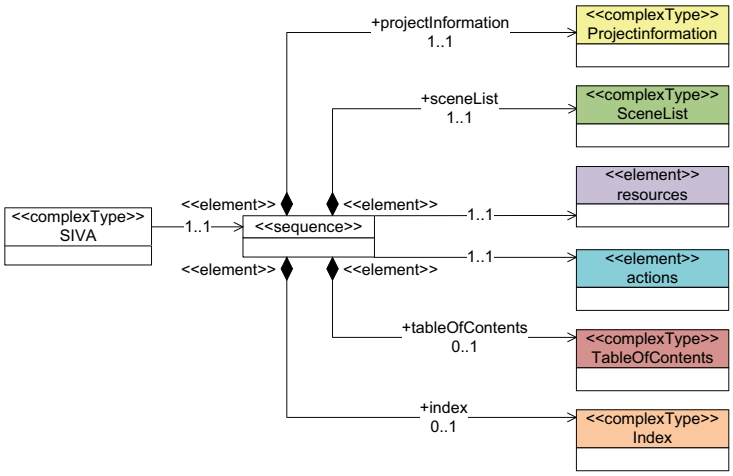


Figure 3.4.: First layer of the SIVA XSD with the six parts project information, scene list, resources, actions, table of contents, and index.

The scene graph (scene list) is explained after the resources and the actions part because both are referenced by the scene list. The structure of the XML file is illustrated for a walk through a house[3]. All line-numbers in this section refer to Listing 3.1 which is explained in the following subsections. We first describe the XSD with UML diagrams which are designed as described by Carlson [Car01d; Car01a; Car01b; Car01c] and Provost [Pro02]. After that, the corresponding lines in the XML file (Listing 3.1) are explained.

[3]The XSD file can be downloaded from `http://siva.uni-passau.de/sites/default/files/downloads/sivaPlayer.xsd`, the example used in this work can be downloaded from `http://siva.uni-passau.de/sites/default/files/downloads/export.xml`. (accessed April 26, 2014)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<siva xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="sivaPlayer.xsd">
<projectInformation>
  <languages defaultLangCode="de-de">
    <language langCode="de-de"/>
    <language langCode="en-us"/>
  </languages>
  <settings name="startmode" value="full"/>
  <settings name="size_width" value="800"/>
  <settings name="size_height" value="600"/>
  <settings name="area_left_width" value="0.2"/>
  <settings name="area_top_height" value="0.2"/>
  <settings name="area_bottom_height" value="0.0"/>
  <settings name="area_right_width" value="0.2"/>
  <projectResources REFactionID="s-NAPic_1"/>
  <projectResources REFactionID="s-NARtxt_1"/>
</projectInformation>
<sceneList REFsceneIDstart="NSc_1">
  <scene REFresID="v_Sc_1" name="Entrance" sceneID="NSc_1" xPos="0.063" yPos="0.101">
    <storyBoard REFactionIDend="select-NSel_1">
      <trigger REFactionID="s-NAa_1" endTime="00:00:07.211"
            startTime="00:00:03.230" triggerID="t-NAa_1"/>
      <trigger REFactionID="s-NAa_2" endTime="00:00:13.440"
            startTime="00:00:07.236" triggerID="t-NAa_2"/>
      <trigger REFactionID="s-NAPic_2" endTime="00:00:09.304"
            startTime="00:00:00.000" triggerID="t-NAPic_2"/>
      <trigger REFactionID="s-NARtxt_2" endTime="00:00:03.799"
            startTime="00:00:00.000" triggerID="t-NARtxt_2"/>
      <trigger REFactionID="s-NARtxt_3" endTime="00:00:13.440"
            startTime="00:00:04.781" triggerID="t-NARtxt_3"/>
      <trigger REFactionID="s-NARtxt_4" endTime="00:00:10.183"
            startTime="00:00:01.912" triggerID="t-NARtxt_4"/>
    </storyBoard>
  </scene>
  <!-- ... -->
  <scene REFresID="v_Sc_4" name="Exit" sceneID="NSc_9" xPos="0.31" yPos="0.43">
    <storyBoard REFactionIDend="end-siva">
      <trigger REFactionID="s-NAPic_4" endTime="00:00:13.640"
            startTime="00:00:00.000" triggerID="t-NAPic_4"/>
    </storyBoard>
  </scene>
  <!-- ... -->
  <scene REFresID="v_Sc_5" name="Living room" sceneID="NSc_6" xPos="0.525" yPos="0.652">
    <storyBoard REFactionIDend="load-NSc_10">
      <trigger REFactionID="s-NAPic_5" endTime="00:00:20.440"
            startTime="00:00:00.000" triggerID="t-NAPic_5"/>
    </storyBoard>
  </scene>
</sceneList>
<resources>
  <videoStream resID="v_Sc_1">
    <content href="videos/v_Sc_1-de_DE.flv" langCode="de-de"/>
  </videoStream>
  <!-- ... -->
  <audioStream resID="a_NAa_1">
    <content href="audios/Audio_1-de_DE.mp3" langCode="de-de"/>
  </audioStream>
  <!-- ... -->
  <richPage resID="rp_NARtxt_1">
    <content href="richpages/RT_1-de_DE.html" langCode="de-de"/>
  </richPage>
  <!-- ... -->
  <image resID="i_NAPic_1">
    <content href="pix/Pic_1-de_DE.jpg" langCode="de-de"/>
  </image>
  <!-- ... -->
  <label resID="l_t_TI_1">
```

```
69      <content langCode="de-de">House</content>
70    </label>
71    <!-- ... -->
72  </resources>
73  <actions>
74    <showImage REFresID="i_NAPic_1" actionID="s-NAPic_1" pauseVideo="false">
75      <area screenArea="right"/>
76    </showImage>
77    <!-- ... -->
78    <showRichPage REFresID="rp_NARtxt_1" actionID="s-NARtxt_1" pauseVideo="false">
79      <area screenArea="right"/>
80    </showRichPage>
81    <!-- ... -->
82    <showRichPage REFresID="rp_NARtxt_4" actionID="s-NARtxt_4" pauseVideo="false">
83      <path>
84        <point time="00:00:09.618" xPos="0.248" xSize="0.497" yPos="0.326" ySize="0.32"/>
85        <point time="00:00:10.138" xPos="0.32" xSize="0.497" yPos="0.27" ySize="0.32"/>
86      </path>
87    </showRichPage>
88    <!-- ... -->
89    <playAudio REFresID="a_NAa_1" actionID="s-NAa_1" muteVideo="true" pauseVideo="false"/>
90    <!-- ... -->
91    <showSelectionControl REFcontrolIDdefault="NSelCtrl_1"
92          REFresID="l_t_NSel_1" actionID="select-NSel_1" timeout="00:00:00" type="default">
93      <path>
94        <point time="00:00:00.000" xSize="-1.0" ySize="-1.0"/>
95      </path>
96      <controls REFactionID="load-NSc_2" REFresID="l_t_NSelCtrl_1" controlID="NSelCtrl_1"/>
97      <controls REFactionID="load-NSc_3" REFresID="l_t_NSelCtrl_2" controlID="NSelCtrl_2"/>
98    </showSelectionControl>
99    <!-- ... -->
100   <showMarkControl REFactionID="i_NAPic_6" actionID="SMC_1" duration="00:00:15.000">
101     <ellipse>
102       <ellipsePath time="00:00:01.010" xPos="0.013333334"
103             yPos="0.024" lengthA="0.27666667" lengthB="0.192"/>
104       <!-- ... -->
105       <ellipsePath time="00:00:24.120" xPos="0.8333334"
106             yPos="0.0" lengthA="0.14333333" lengthB="0.1"/>
107     </ellipse>
108   </showMarkControl>
109   <!-- ... -->
110   <loadVideoScene REFsceneID="NSc_1" actionID="load-NSc_1"/>
111   <!-- ... -->
112   <endSiva actionID="end-siva"/>
113 </actions>
114 </siva>
```

Listing 3.1: Example XML

3.1.2.1. Project Information

The complex type ProjectInformation consists of a sequence of the following elements: one `languages` element and one or more `settings`, `projectResources`, and `resourceSettings` elements. A UML diagram of the structure of this complex type can be found in Figure 3.5. The `languages` element is of the complex type Languages which contains a `defaultLangCode` attribute describing the language at player start-up. The complex type Languages includes a sequence of `language` elements of the complex type LangCode which embodies the attribute `langCode`. The `projectResources` element is of the complex type ProjectRessource, which contains a `REFactionID` attribute. This `REFactionID` points to an action which causes the display of a resource. The `settings` element is of the complex type Settings. It encloses the attributes `name` and `value` which contain impor-

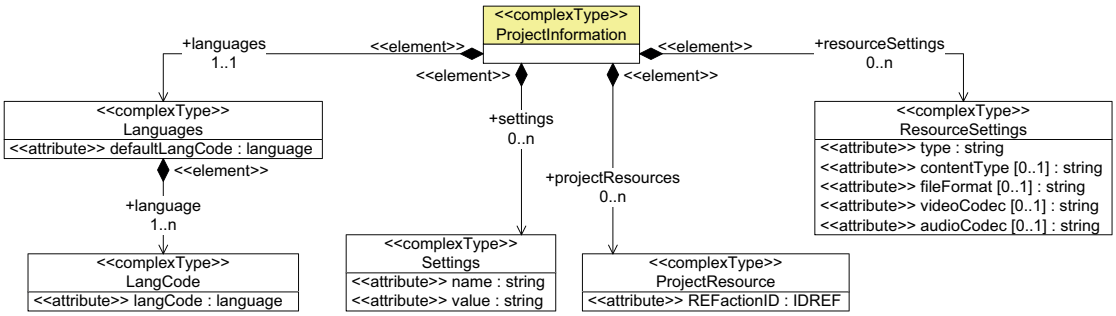


Figure 3.5.: The complex type "ProjectInformation".

tant configuration settings for the player. The `resourceSettings` element is of the type ResourceSettings. This complex type contains the five attributes `type`, `contentType` (optional), `fileFormat` (optional), `videoCodec` (optional), and `audioCodec`(optional).

An XML example of the project information part can be found in lines 4-18 in Listing 3.1. It defines the languages available for the annotated interactive non-linear video (lines 5-8). Different settings are possible according to the capabilities of the player implementation. They are described in settings element `<settings name="..." value="..."/>` (lines 9-15) where a `value` can be assigned to a `name`. The settings in the example are specified for the SIVA Desktop Player. Four static areas are defined in this player and can easily be set to a size (lines 12-15). The project resources determined in lines 16-17 are shown during the whole annotated interactive non-linear video in the right annotation area. Project resources usually do not pause the main video and in this case they do not mute it.

3.1.2.2. Resources

The `resources` element contains the subelements `videoStream`, `audioStream`, `image`, `richPage`, `plainText`, `subTitle`, and `label` in any order. We used inheritance to define the different types of resources. All types of resources are derived from the complex type Resource which solely embodies a `resID` attribute. LinkedRessource and PlainTextRessource form the second layer of the inheritance hierarchy. A PlainTextRessource includes a `content` attribute and an attribute for the language code (`langCode`). The elements `plainText`, `label`, and `subTitle` are from this complex type. A LinkedRessource contains one or more `content` elements of the complex type Content which consists of a `href` and a `langCode` attribute. The elements `image` and `richPage` are from this complex type. The complex types VideoStream and AudioStream form the third layer of the inheritance hierarchy and they are derived from LinkedRessource. The VideoStream comprises the attributes `audioCodec` (optional), `containerFormat` (optional), and `videoCodec` (optional). The AudioStream only has an optional `audioCodec` attribute. See Figure 3.6 for an overview.

In the XML file, resources state the content of an annotation or a scene (lines 51-72 in Listing 3.1). It is possible to compile them in different languages. Resources are displayed by the action function based on the ID. Labels and subtitles are defined inline, because they mostly consist of only a few words and are commonly styled by a player. Examples of an inline defi-

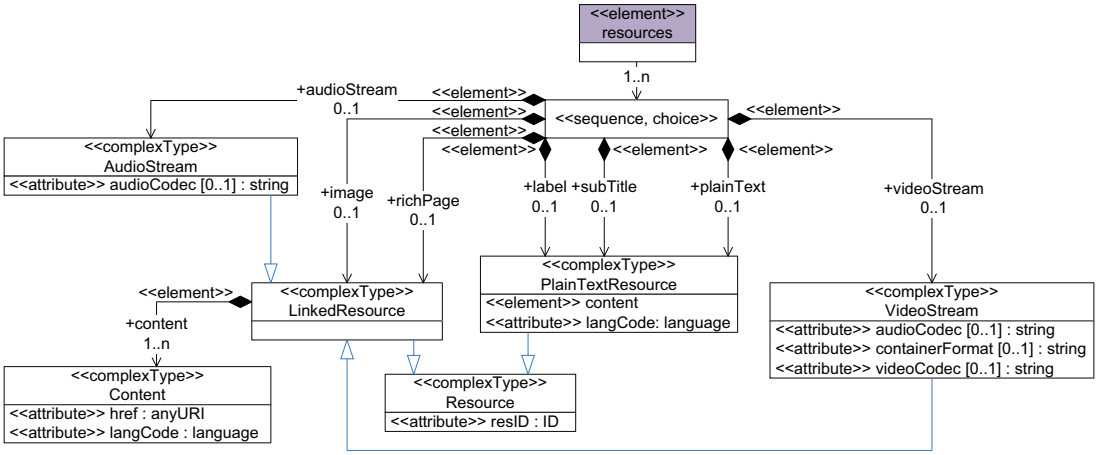


Figure 3.6.: The element “resources”.

nition of resources can be found in line 69. All other resources - images, rich texts, video, and audio files - are defined as links in the XML file, see lines 53, 57, 61, and 65.

3.1.2.3. Actions

The `actions` element contains a sequence selection of interactive actions (described later in this section) and the following static action elements: `loadVideoScene`, `playAudio`, `showSubtitle`, `showPlainText`, `showImage`, `showRichPage`, `showVideo`, and `endSiva`. The UML schema of the static actions is illustrated in Figure 3.7. As described in the resources subsection, inheritance is used for actions, too. The complex supertype of all actions showing resources is Action. It contains the attributes `actionID`, `pauseVideo`, and `showModal`. The last two are set to false in the standard case and are optional. Because nothing is displayed invoking the `playAudio` action, ActionPlayAudio is derived from Action directly. Furthermore it is extended by the attributes `REFresID` and `muteVideo` (optional). ActionShowSubtitle is derived from Action directly, as well, because the subtitles are always shown at a fixed position. It is extended only by a `REFresID` attribute. The other actions which show a resource need spatial information for display. This information is set in an `area` element or by defining a path in the `path` element in the complex type ActionShow. The `area` element is from the complex type Area, which includes a `screenArea` attribute, which is filled by the ScreenArea enumeration. The `path` element is of the complex type Path which contains a sequence of `point` elements. A `point` element is of the complex type Point, which embodies a `time` attribute and the attribute group SubWindow. The attribute group SubWindow then again contains the attribute groups SubWindowSize (attributes: `xSize`, `ySize`) and Position (attributes: `xPos` (optional), `yPos` (optional)). ActionShowAnnotation, ActionShowStream, and ActionShowAnnotationGallery are derived from ActionShow. ActionShowStream contains the attribute `REFresID` pointing on a videoStream resource and an optional `muteVideo` attribute. The action `showVideo` is of this complex type. The actions `showPlainText`, `showImage`, and `showRichPage` are of the complex type ActionShowAnnotation which contains a `REFresID` attribute. The action `showImages` is of the complex type ActionShowAnnotationGallery which includes an optional `columnCount` attribute. It includes a `galleryResources` element of the com-

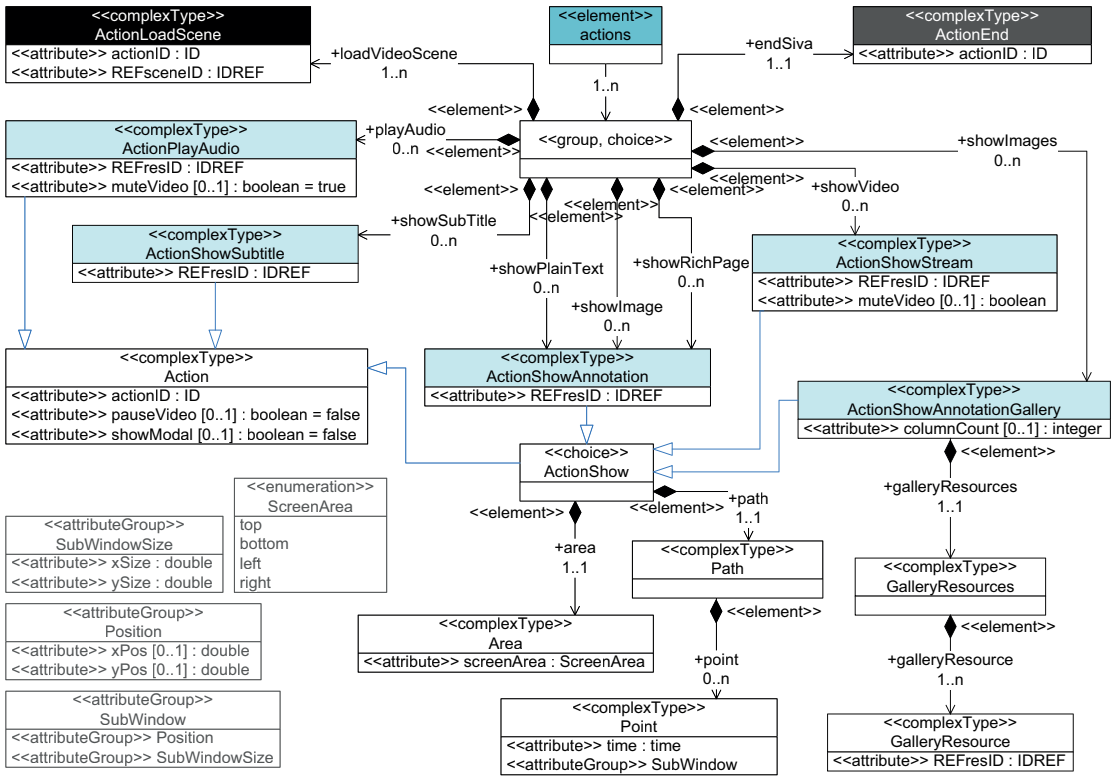


Figure 3.7.: The complex types for the static actions.

plex type GalleryResources which then again consists of one or more `galleryResource` elements of the complex type GalleryResource. The attribute `REFresID` is the only content of GalleryResource. The elements `loadVideoScene` and `endSiva` occupy a special position. The action `loadVideoScene` is of the complex type ActionLoadScene with attributes `actionID` and `REFseceneID`. It is used to load a scene. The `endSiva` action is of the complex type ActionEnd, enclosing an `actionID` as the only attribute. This action is invoked at the end of a video and stops the player.

The interactive parts of the `actions` element are the `showSelectionControl` and `showMarkControl` element. The complex type ActionShowPanel is derived by the complex type Action and contains the optional attribute `REFactionID`. Spatial information for display is modeled in a choice of an `area` element or a `path` element as described in the previous paragraph. ActionShowSelectionControl is derived by ActionShowPanel and contains the attributes `REFcontrolIDdefault` (optional), `timeout` (optional), and `type`. The `type` attribute is set by the enumeration ControlType thereby. The `showSelectionControl` action is of the complex type ActionShowSelectionControl and contains a sequence of `controls`. These are of the complex type Control, which contains a `controlID`, a `REFactionID`, a `REFresID`, and an optional `REFresIDsec` attribute. The latter ones allow the creation of buttons with icons and labels. The complex type ActionShowMark is derived by Action, too, and encloses the attributes `REFactionID`, `duration` (optional), and `style` (optional) in addition. Furthermore, an ActionShowMark contains either a `polygon` element, a `button` element, or an `ellipse` element. The `polygon` element is of the complex type Polygon which contains a sequence of `polygonalChain` elements of the complex type PolygonalChain with a `time` attribute. The PolygonalChain then again includes a sequence of

vertices elements of the complex type Vertex including the attribute group Position, which consists of the attributes xPos and yPos. The button element is of the complex type Button, which embodies an optional REFactionID attribute and a sequence of buttonPath elements. The buttonPath elements are of the complex type ButtonPathElement, with a time attribute and the attribute group Position. The ellipse element is of the complex type Ellipse which contains a sequence of ellipsePath elements. The ellipsePath elements are of the complex type EllipsePathElement and include the attributes time, lengthA, and lengthB as well as the attribute group Position. The UML diagram of interactive actions is illustrated in Figure 3.8.

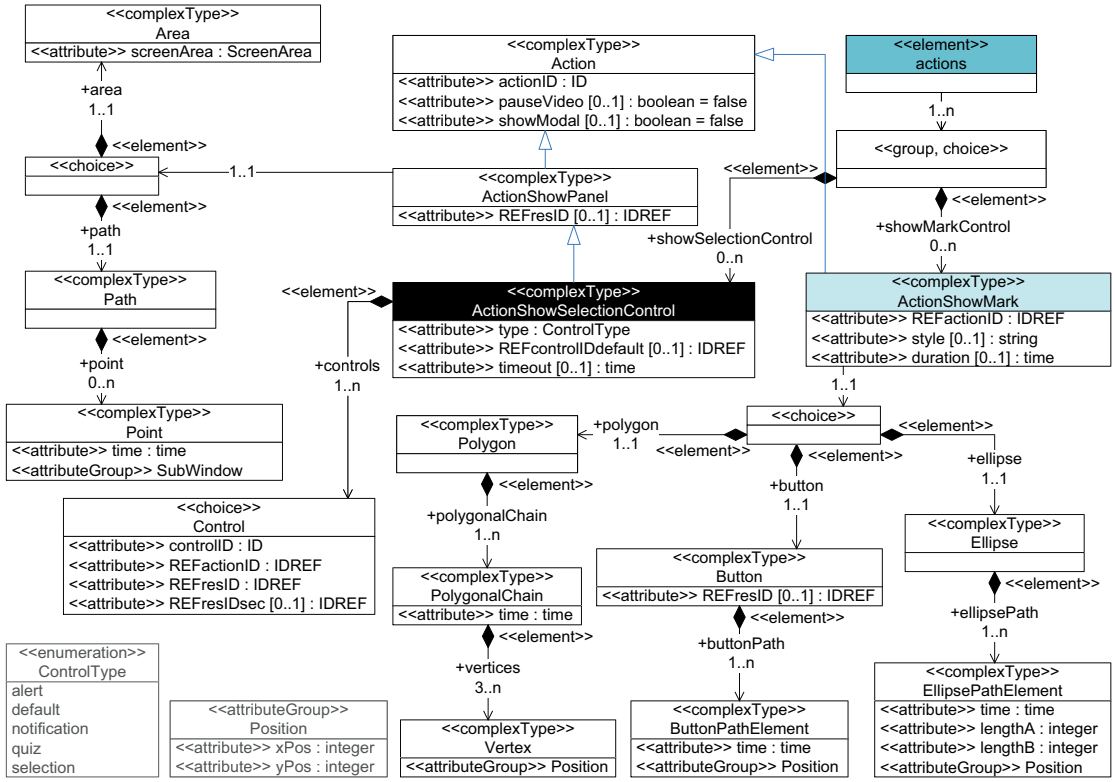


Figure 3.8.: The complex types for the dynamic actions.

Actions are triggered at certain points in time in the video or by a user interaction. They are described in the XML file (Listing 3.1) in lines 73-113. Actions cause the display of a resource, a selection panel, a marked object in the video, load a scene, or indicate the end of a video.

- *Display of additional information ("static annotations")*
  All actions causing the display of an annotation have a reference REFresID to one of the resources (lines 74, 78, 82, and 89). No action is able to pause the video when it is displayed, which is stated in the pauseVideo attribute in lines 74, 78, 82, and 89. Actions starting a video or an audio file are able to mute all other annotations and the main video, as defined in the optional muteVideo attribute. The playAudio element (line 89) is the only one which mutes the video in the example. The actions have a positioning information. They are either displayed in one of the areas (lines 75 and 79) defined in the project information part of the XML file (lines 12-15), or they are displayed as an overlay having a time-based position and size information (lines 83-86).

- *Action to load a new scene*
  A new scene can be loaded with a `loadVideoScene`-action (line 110). The element consists of a reference to a scene from the scene list (`REFsceneID`) and an `actionID`.

- *Button panel*
  A button panel is defined in the `showSelectionControl` element in lines 91-98. It consists of two or more buttons (two in our example) as stated in the `controls` elements. Each button has a `controlID` for a unique identification, a reference `REFresID` to a label, and a reference to load a scene or another `showSelectionControl` element (not in this example), which is loaded after a click on the button. The button panel also has a reference to a label `REFresID` and an `actionID` to be referenced at the end of a scene. One of the `controls` can be defined as default control in `REFcontrolIDdefault` (line 91). It is selected after the period of time set in the `timeout` attribute (line 92). A positioning information is set similar to the position of an action displaying a resource. The button panel will be displayed as a centered overlay over the video area, because of the negative coordinates in line 94.

- *Marked object in the video*
  Clickable objects are expressed by a `showMarkControl`-action as shown in lines 100-108. After the user has clicked on a marked object in the video, a referenced resource by the action set in the `REFactionID` attribute in line 100 is loaded. An object can be marked with an elliptic outline, a polygon outline, or a labeled button which can be placed near the object. The example shows the definition of an elliptic outline (lines 101-107). A `duration` is set to determine how long the referenced annotation has to be shown. In our example it will be shown for 15 seconds (line 100). This cannot be solved by a trigger, because it depends on the time of the user interaction.

3.1.2.4. Scene List

The most important part of the XML file is the scene list (see lines 19-50 in Listing 3.1). It links scenes with the video contents and defines the structure of the whole video (a scene graph). Triggers for displaying and hiding of annotations are set with each scene. The complex type SceneList (see Figure 3.9) with the attribute `REFactionIDstart` consists of a sequence of `scene` elements. A `scene` element is of the complex type Scene which includes the attributes `sceneID`, `REFresID`, `REFresIDname` (optional), and `name` (optional) as well as the attribute groups RangeOfFrames (attributes `startTime` and `endTime`) and Position (attributes `xPos` and `xPos`). The complex type Scene contains a `storyBoard` element of the complex type Storyboard. The complex type Storyboard encloses a `REFactionIDend` attribute and a sequence of `trigger` elements. These `trigger` elements are of the complex type Trigger, which embodies the attributes `triggerID`, `timeout` (optional), and `REFactionID` as well as the attribute group RangeOfFrames.

- *Action at the end of a scene*
  The start scene of the video is defined as `<sceneList REFsceneIDstart="NSc_1">`. Each scene has a storyboard which defines what happens during the scene and when the scene is over. This is defined in `REFactionIDend` in the `storyBoard` element in lines 21, 38, and 45. The example shows all possible end actions after a scene. A linear transition between two scenes is defined with `<storyBoard REFactionIDend="load-NSc_10">` in line 45. Line 38, `<storyBoard`

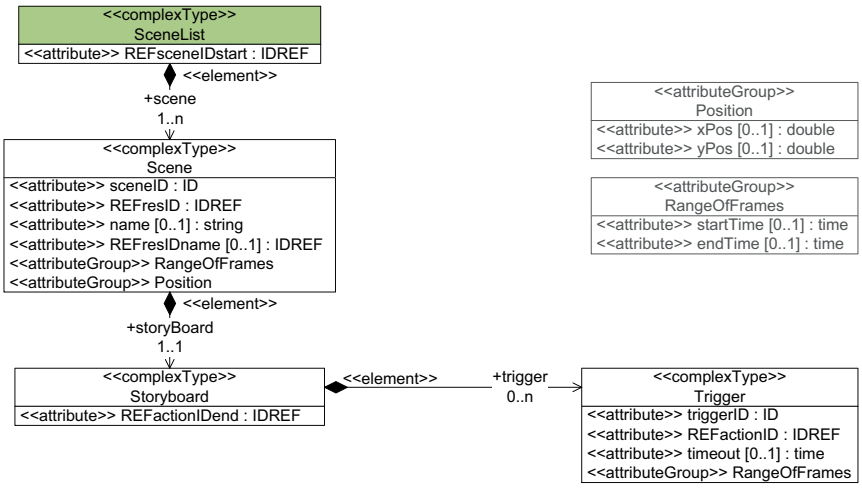


Figure 3.9.: The complex type “SceneList”.

`REFactionIDend="end-siva">`, references the end action of the video. A button panel is shown after the first scene as defined in line 21.

- *Triggers during a scene*
  The points in time when annotations are shown and hidden are also defined in the storyboard of a scene. A `trigger` element (see lines 22-33, 39-40, and 46-47) consists of an action to show an annotation, the start time when the action is performed, the end time when the annotation is hidden again, and an ID of the trigger, for example:

```
<trigger REFactionID="s-NARtxt_3" endTime="00:00:13.440"
    startTime="00:00:04.781" triggerID="t-NARtxt_3"/>
```

3.1.2.5. Table of Contents

The complex type TableOfContents contains the optional `REFresID` attribute in order to point on a label as well as as sequence of `contents` elements. A UML diagram of the structure can be found in Figure 3.10. The `contents` element is of the complex type ContentsNode, which includes the attributes `contentsNodeID`, `REFresID`, and `REFactionID` (optional). Furthermore, the `contents` element contains a sequence of `adjacencyRefListNode` elements. These are of the complex type AdjacencyRefListNode which encloses a `REFcontentsNodeID` attribute. The complex type TableOfContents furthermore embodies a choice of an `area` or a `path` element. The structure of these elements is the same as described in the actions subsection.

The table of contents is defined in the form of an adjacency list in the XML file. An entry with no link to a scene is defined with a reference to a label `REFresID` and a `contentsNodeID`. An entry with a link to a scene is defined in a similar way: `<contents REFactionID="load-NSc_4" REFresID="l_t_TI_6" contentsNodeID="TI_6"/>`. Only the reference to an action-ID is added to refer to the scene which has to be loaded. If an entry is a leaf of the tree structure, no further lines are added. Sub-nodes are added with a reference to another node if the entry is an internal node.

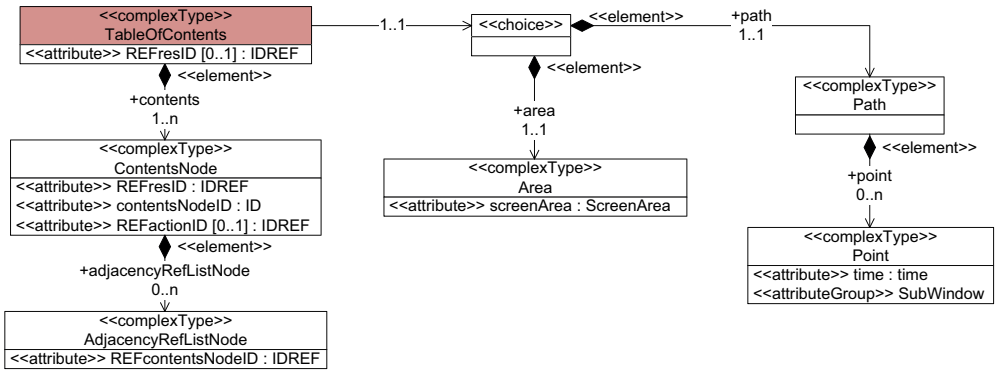


Figure 3.10.: The complex type "TableOfContents".

3.1.2.6. Keyword Search

The complex type Index consists of a sequence of `keyword` elements. A UML diagram of the structure is illustrated in Figure 3.11. The `keyword` element is of the complex type Keyword, which contains the optional attributes `word` and `REFresID` as well as a sequence of `scene` elements. The `scene` element is of the complex type SceneREF and has the attributes `REFsceneID`, `REFTriggerID` (optional), and `ressourceType` (optional). The `ressourceType` is set by the enumeration ResourceType.

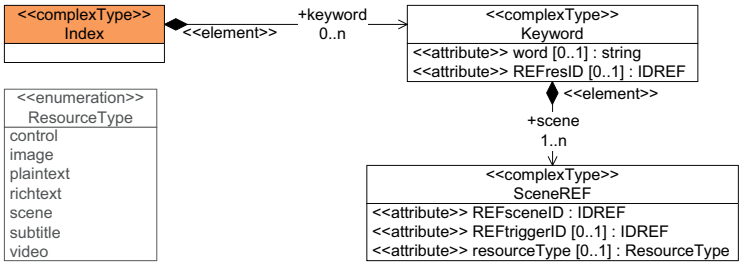


Figure 3.11.: The complex type "Index".

Keywords listed in the `index` element in the XML file are arranged in a list structure where every entry of the main list has a sub-list. The main list consists of keyword elements `<keyword word=...>`. Each of the keywords has a sub-list with elements of the video matching the keyword. The scene is loaded and played from the beginning, if the user selects the entry in the search results page. If an annotation matches the keyword, the player starts at the point in time of a scene where the annotation is displayed. (Parts of this section (3.1 SIVA XML Schema: Modeling of Annotated Interactive Non-linear Video Behavior) were taken and adapted from our previous works [MK12] and [Mei+10b].)

3.2. SIVA Producer: Management of Interactivity, Non-linearity, and Annotations in an Authoring Tool

The authoring process of annotated interactive non-linear videos is more difficult than the process for traditional videos without interactivity and non-linearity. More media files and an overview of the non-linear structure are needed to produce a consistent and appealing presentation. These media files need to be managed in an authoring tool, which requires a good comprehensibility to promote a fast learning of the software functions as well as a quick editing of the video projects. Ogawa et al. [Oga+92] describe four steps to accomplish that goal. The first step is to define a global structure of the material, then each module (in our case a video) is planned in detail. After that, the contents are specified and finally a presentation style is designed. When this planning phase is over, the project can be implemented with a software. Hofmann, Hollender, and Fellner describe "a set of requirements [for a software] that is based on the exemplified tasks and processes of the video annotation workflow" [HHF09]. They identify the categories "configuration, segmentation, annotation, exploration, and externalization" [HHF09]. From these arise requirements for an annotation software. They name "workflow-related" and "collaboration" as further categories. Some of the hints on implementing the requirements in a software can be used for our authoring tool. Furthermore, the implementation process for annotated interactive non-linear videos can be simplified if the used software provides as much user support and is as user friendly as possible. Usability tests were performed during the whole development process. Results are integrated into the software for an improved usability. Single tests and their results are described in [MGK12; Mei+11a; Mei+12a; Mei+12b; Mei+12c; Kuc13].

The SIVA Producer provides all functions to manage the creation of an annotated interactive non-linear video. It is implemented as an ERCP[4] application using the MVC pattern. The ERCP provides an operating system like look and feel and allows to arrange editors and views in a predefined way which can be changed by the user. Different (Java-)libraries and extensions were evaluated for their usefulness in the authoring tool in Meixner et al. [Mei+11b] and Fichtelmann [Fic13]. The SIVA Producer provides the following functions:

- Configuration of settings for the player,
- import of media files into the project,
- video editing,
- creation of non-linear video structure with a scene graph,
- creation selection panels,
- enrichment of the video content with additional multimedia annotations,
- creation of a table of contents,
- tagging of annotations and scenes with keywords, and
- export to different formats.

[4] Eclipse Rich Client Platform - `http://wiki.eclipse.org/index.php/Rich_Client_Platform` (accessed April 26, 2014)

Two superior modules of our software are the project management and the video editing part. The **project management** part of the SIVA Producer incorporates standard functions known from other tools. It consists of an import part, an export part, and a settings part. The import part is used to insert all media and pre-built contents like images or texts into a *media repository* (area 3 in Figure 3.12). The media repository is based on a tree structure. The image files are provided with a thumbnail button. It is possible to add a descriptive file name for all files. All media files are arranged in a grouped way. They can be dragged and dropped to the annotation editor or they can be modified with basic built-in editors for the different types of annotations. For example, the image editor provides functions to add rectangles or texts, and to mark objects in an image. A *settings wizard* provides assistance in configuring the look of the player. The size of available annotation areas is set and a color scheme can be applied. Our software allows to save and *export* projects, either for passing on to another author or to a web server. The export for the web server produces a folder structure containing the player files, the media files, and the XML file. All media files are converted to a format that is suitable for the respective platform. The authoring software furthermore has a **video editing** function. It provides a manual and an automated way to define scenes. A timeline-based editor was implemented to *cut whole videos into scenes manually*. The editor has a video preview, an overview which contains the already completed scenes, and an area where the boundaries of a scene can be defined (by sliders on a timeline or by entering the time or frame numbers manually). The structure of the editor is similar to that of the annotation editor in Figure 3.14. An *automated scene detection* as described in [ZMK14] can be used to create scenes, too. The user is able to correct the output of an automated shot detection manually. The automated scene detection finds scenes which can again be changed by the user. After finishing this process, the scenes are exported to the scene repository (area 2 in Figure 3.12) from where they can be inserted into the scene graph.

We decided to integrate the graph-based paradigm in our authoring tool because it is a good way to show cycles, forks, and parallel courses of scenes. Furthermore, the timeline-based paradigm is well suited for our scene editor and the annotation editor, because both editors deal with one single linear video or scene. No cycles or forks need to be modeled. The annotation editor needs an overview of one scene and the annotations which are displayed and hidden during that scene, which is provided by the timeline-based paradigm flawlessly. The scene editor has to show the results of editing a video and building scenes. The timeline-based pattern shows all defined scenes and provides an overview on parts of a video which is contained in more than one scene. Good ideas to improve usability are repositories with a folder structure as described in HyperProp [SRMS00], the display of information about size and position of objects on the video canvas as depicted in HVLG [Hun97], and the hotspot editor for the creation of polygonal overlays on the video canvas as presented for HyPE [HH06].

The remainder of this section gives an overview on how our authoring tool implements the construction of interactivity, non-linearity, and additional information. Functions of these categories are described in more detail as follows.

3.2.1. Non-linearity

Each non-linear navigation in the video has to be defined in the authoring tool by defining structures and entry points for video scenes. The scene graph is used to create basic navigation. Extended navigation is implemented by a table of contents and a keyword search. The

keyword search is realized by using annotations and scene names which are linked at player side to create timestamps at which a scene starts playing. It is described in Subsection 3.2.2 for this reason.

Scene Graph: A **scene graph** is used to model the non-linear structure of the video. The graph-based paradigm is implemented as a directed graph model according to [BH05]. Tree structure or simpler scene lists cannot be used in our authoring tool due to the cycles that may be defined. The scene graph defines the order of scenes, forks in the flow of the whole video and points where user interaction is required. To provide a better overview of the graph and the annotations linked to the scenes, three different **semantic zoom levels** and a **semantic fish-eye** are implemented. A **geometric (standard) zoom** allows the user to change the size of the whole graph. The zooming functionality was evaluated for its usefulness in Meixner, Grill, and Kosch [MGK12]. For further descriptions of this feature see [Gri11]. Figure 3.12 shows the scene graph in semantic zoom level one. A snippet of the graph in semantic zoom levels two and three can be found in Figure 3.13. Scenes are added to the graph by drag and drop from the scene repository. They are linked with the arrow tool from the toolbox. The structure of the graph has to follow several rules: It must have a start scene and at least one edge to the end element. Transitions with Boolean expressions are created by fork nodes realized as gray diamonds with dedicated rectangles. Area 4 of Figure 3.12 shows the currently displayed part of the graph editor. An information outline about the currently selected element is shown in Figure 3.12 area 5. A click on the orange buttons on a scene rectangle opens the annotation editor.

Table of Contents: A table of contents can be created in a two-tier editor (see Figure 3.14, right). The right side of the editor shows all scenes that are added to the scene graph so far. Only these scenes are part of the annotated interactive non-linear video and will be exported to the XML file consequently. These scenes can be linked with the table of contents. The left part of the editor allows the author to create the entries of the table of contents. All entries can be created, positioned, and edited. Scenes are linked to an entry by dragging them from the list on the right side and dropping them on an entry on the left side.

3.2.2. Annotations

Additional information and annotations are added to a video with two different principles. Scenes may contain additional information and they can be annotated with keywords for the search function in the player. Annotations themselves may also be annotated with keywords.

Annotation Editors: The **annotation editor** (Figure 3.14, left) is realized with the timeline-based paradigm. The intention of doing this was to give an overview of the temporal sequence of annotations in a scene. Annotations are only valid during one scene, except global annotations. All kinds of annotation can be created and edited there. It is possible to add an annotation to more than one scene. Files which were imported into the media repository can be added as annotation content by drag and drop. The time when the annotation is displayed and hidden can be set in this editor via mouse click or by filling in the exact times. It is possible to determine one area where the annotation is displayed in the player. It is also feasible to define a path on which the annotation moves along in an overlay over the video. Furthermore, the main video can be paused

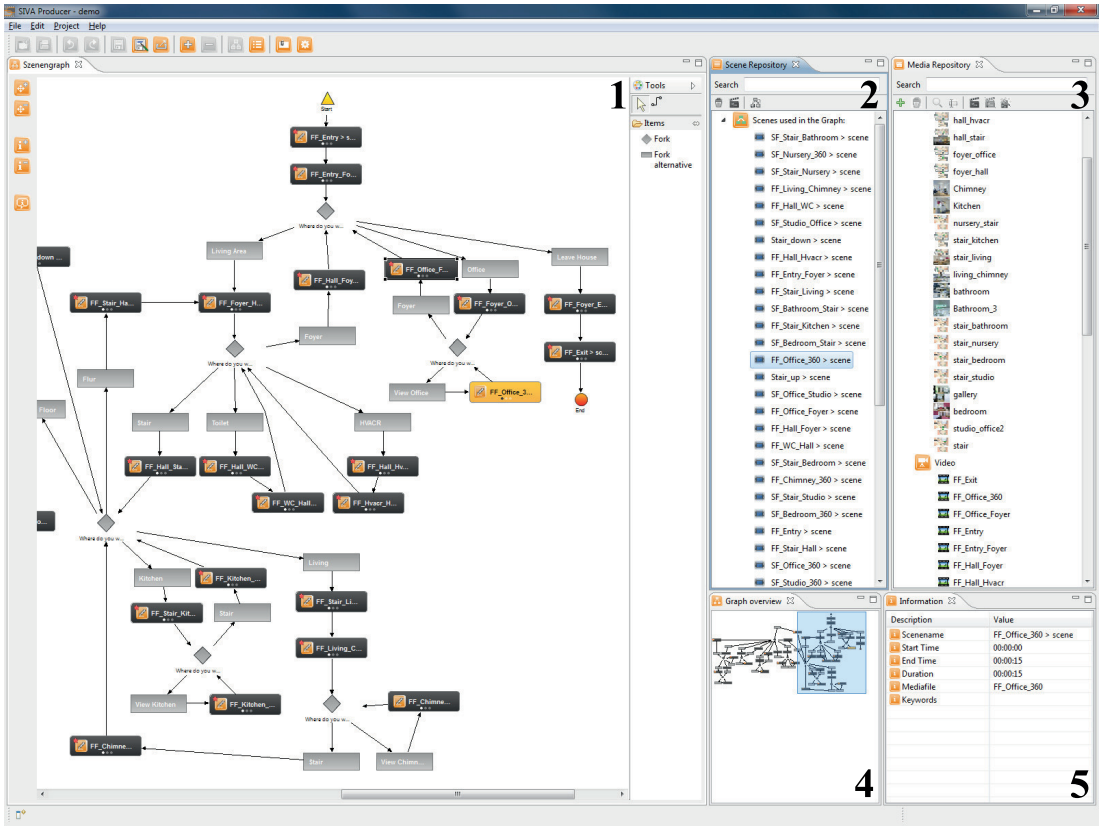


Figure 3.12.: Screenshot of the SIVA Producer: (1) scene graph, (2) scene repository, (3) media repository, (4) graph overview, (5) information area.

Figure 3.13.: Presentation of scenes in the scene graph: fish-eye zoom (left), zoom level two with a video preview and the number of the different annotations (center), zoom level three with a video preview, information about the video, a list of annotations and their position (right).

when the annotation is displayed. Thereby, the main video and all other annotations can be muted. An annotation which is displayed during the whole duration of the scene is created by dragging and dropping the content of the annotation on a scene in the scene graph. **Global annotations** which are displayed during the whole playback of the video are created in a separate editor because no timing information has to be set for them. Consisting of two parts, one part of the editor gives an overview of existing global annotations. The second part enables the user to add content and set the (fixed) position of the annotation.

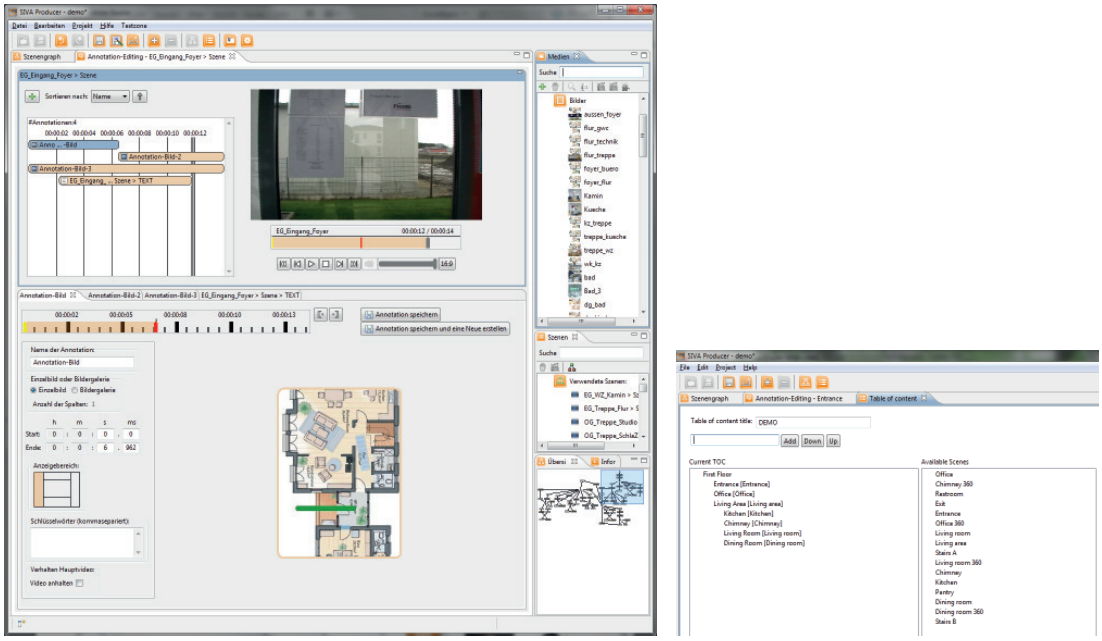


Figure 3.14.: Screenshots of the SIVA Producer: the annotation editor with a timeline view, a video preview, and an area for annotation content and settings (left), table of contents editor with a table of contents in the left part and available scenes in the right part (right).

Definition of Keywords: Keywords can be assigned to scenes and annotations. For that reason, we provide a text field where the user can set a list of comma-separated keywords (see Figure 3.14, left) for each scene and each annotation in the annotated interactive non-linear video. The text fields are integrated in the scene and the annotation editor. The export function analyzes the comma-separated list and adds the keywords to scenes and annotations in the index section of the XML file.

3.2.3. Interactivity

Each click in the player is a form of interaction with the video or its additional information, but only a part of this interactivity needs to be defined in the authoring tool. The appearance and the movement of hotspots are configured in an extension of the annotation editor. Selection elements in the scene graph like a button panel is implemented in the scene graph.

Hotspots: Hotspots are clickable areas in a video. Their appearance and their movements as well as their behavior have to be defined in the authoring tool. An extended annotation

editor (see Figure 3.15) is used to define the behavior after a click on the hotspot. Thereby, our tool allows us to define an additional information which is opened in one of the annotation areas or as an overlay over the video after a click on the hotspot. The editor furthermore defines the shape and the movement of the clickable area of the hotspot. Possible shapes are a button, an ellipse (see Figure 3.15), or a polygon. The button adjusts its form according to the contained text automatically. The number of edges and the shape of the polygon or the length of the axes of the ellipse need to be defined and changed at certain points in time if an object in the video alters its shape. Reshaping is accomplished by dragging one of the anchors of the shape and moving it to the new position. For repositioning, the whole hotspot is moved by clicking and holding the mouse button down in the marked area of the hotspot and moving the shape in the video preview.

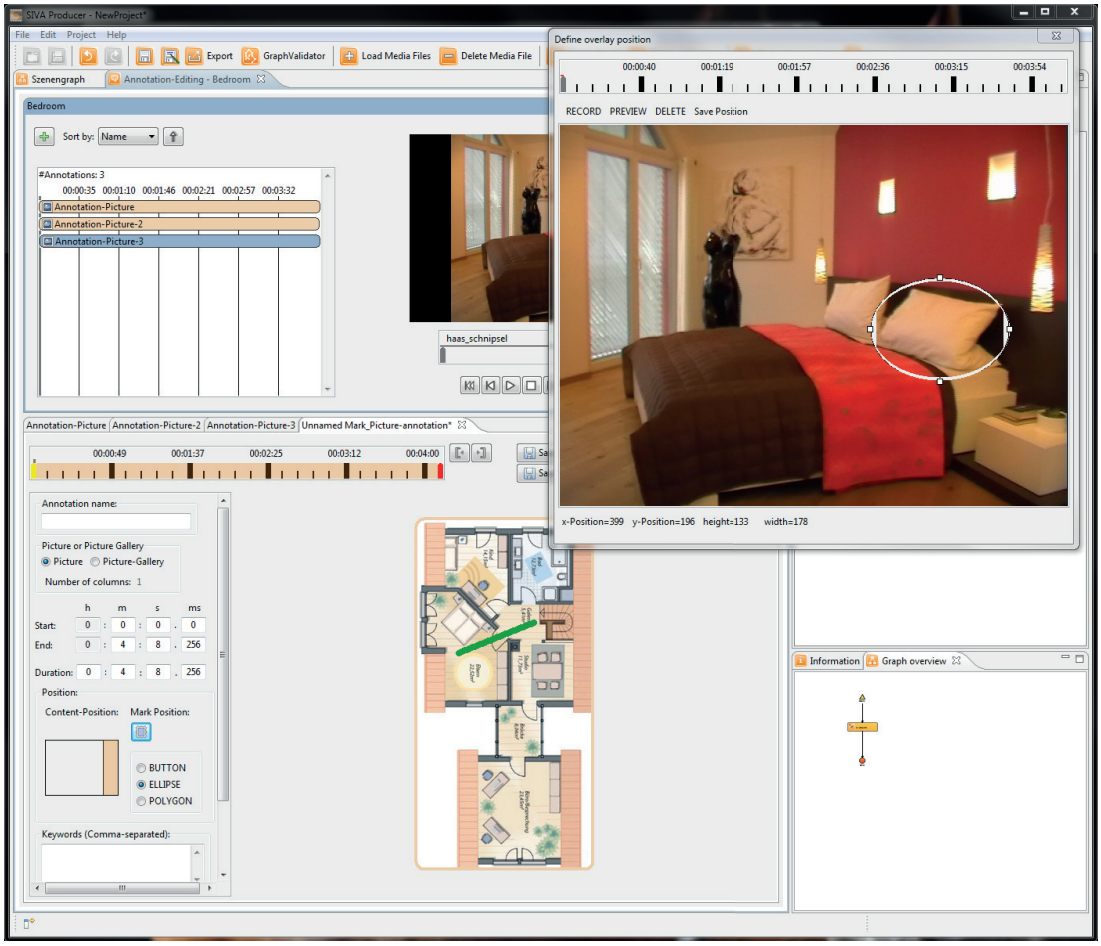


Figure 3.15.: Annotation editor with hotspot editor: definition of an image annotation which is opened by a click on the ellipse.

Selection panel: Selection panels need to be clicked by users to proceed with the video and load the following scene (or the end of the video) in the player. They are defined in the scene graph editor by adding and connecting a fork element (rhombus) with path elements (rectangles). One or more connections into the fork element and one connection from each path element to a scene is necessary to create a valid scene graph. The label of the fork element represents the question/headline of the selection panel in

the player. A path element may have a label, a picture, or both. Each path element is displayed as a button in the players. The elements for the creation of a selection element are inspired by the notation of flowcharts as standardized in [Int85].

(Parts of this section (3.2 SIVA Producer) were taken and adapted from our previous work [Mei+12b].)

3.3. SIVA Player: Realization of Interactivity, Non-linearity, and Annotations during Playback

Interactivity, non-linearity, and annotations need to be handled in the player. The XML file containing content and control information has to be processed. The main video and all additional information like images, audio files, videos, and rich text with links need to be displayed at the correct time. The player furthermore has to respond properly on user input. Depending on the display size, the presentation of the contents as well as the extent of interaction may vary.

We think that a player for annotated interactive non-linear videos should provide a maximum degree of flexibility in arranging annotations with the main video to make authoring a not too complex task. For that reason, four annotation areas, one positioned at each side of the main video area, combined with overlays seem to solve this issue. The implementation of player controls varies substantially throughout the described interfaces. A player for annotated interactive non-linear videos should at least provide standard controls like play, pause, and a volume control, as well as a timeline of some kind.

Five different prototypical players were developed over the course of the different projects. A prototypical test version of the player was implemented in Microsoft Silverlight[5] [Wei10], but not used in the project due to the low distribution of this technology. Widespread player frameworks/software products like Adobe Flash or HTML5 are preferred, because many users are able to watch annotated interactive non-linear videos without being forced to install additional software. The first official version of the SIVA Web Player was implemented in Adobe Flex[6] due to the lack of alternative frameworks at that time (see evaluation in [Mei+11b]). A brief description and a screenshot of this player version can be found in Meixner et al. [Mei+10a]. An evaluation of this player based on design and web usability guidelines [BK04; UU03; Use09] can be found in Meixner et al. [Mei+11a]. Due to various deficiencies of this player, a switch in technology to HTML5 in combination with JavaScript and CSS was made. The **SIVA HTML5 player** was implemented in a layered architecture. Basic libraries and frameworks are MooTools[7] used for communication, Raphaël[8] for creating SVG vector graphics, Data-Driven Documents (D3)[9] for manipulating documents based on data and generate charts as well as tables, and Joose[10] as meta object system. Self created libraries extend these basic frameworks. Using those libraries and frameworks, a template engine, a data visualization library, and data structures and tools for debugging were implemented. The SIVA

[5] `http://www.microsoft.com/silverlight/` (accessed April 26, 2014)
[6] `http://www.adobe.com/products/flex.html` (accessed April 26, 2014)
[7] `http://mootools.net/` (accessed April 26, 2014)
[8] `http://raphaeljs.com/` (accessed April 26, 2014)
[9] `http://d3js.org/` (accessed April 26, 2014)
[10] `http://code.google.com/p/joose-js/` (accessed December 06, 2012)

desktop player furthermore implements the MVC pattern. Our architecture provides a simple interchangeability of single parts as well as an easy expandability. This enables us to exchange the currently used controller to process SMIL files, for example. The current version of the player processes the XML file described in Section 3.1 to extract all necessary control information for playback. For further descriptions of the HTML5 Player see [Den12; Mei+13]. The description of a previous version of the player can be found in [Spe11]. A logging extension is added to the player to collect information about user behavior, terminal device, and data connection. It is implemented as a plug-in with defined interfaces at player side. The server obtains as much data as possible to extend the data collected by the player or to check these data. An extendable graphical analysis framework was implemented for first evaluations. It collects the following information and saves them in the database for easy access during the analysis:

- *Client information* like browser, type of terminal device, user language, and location information,
- *native browser events* like mouse or keyboard input, and changes of the browser window size; most of these events cause changes of state or data,
- *scene and annotation events* like loading a new scene or displaying and hiding of annotations, and
- *error messages* which are used mainly for debugging or might occur in unexpected player events.

With the widespread use of smart phones and the recognition of "mobile phones as new media interfaces" [MR06], two applications, one for Android and one for Apple devices were developed. With HTML5 still in development and only rudimentary implemented in browsers for mobile devices, the high level of interactivity combined with video playback was not realizable with this technique when the development started in 2011. Therefore we decided to implement two prototypical apps, one on the Android platform, the other one for Apple devices. No apps were developed for Windows Phones and Blackberrys because of their limited market presence at that time.

The Android SDK was used for the development of the **SIVID Player (Android player app)**. It provides a high performance because no additional runtime environment is needed and the code can be executed natively. Furthermore, a wide range of codices is supported. The code itself runs in a Dalvik Virtual Machine (DVM) based on the Apache Harmony JVM [BP10, pp. 21]. A message based MVC pattern as described in [Mem10] is used as well as techniques for better code performance like the principle of simplicity, the avoidance of encapsulation, the waiver of getters and setters, and the usage or the avoidance of certain structures or data types (see [And; BP10]). The SIVID Player has to fulfill the same requirements as the web player introduced in [Mei+10a]. The home screen of the application is based on the dashboard-pattern [Ful+10]. It provides six different functions. One is for resuming a video that was displayed before. A function for searching interactive videos on the Web or searching in the local library is implemented. A library provides an overview of the videos that were downloaded before with an intelligent download function. The behavior of the application can be configured in a settings dialog. Information about the application is shown in an info panel. Bookmarks that were set in already watched videos can be shown in a list. If different languages are defined in the XML file, a control is provided to switch between them. Wherever it is possible, GUI-patterns [Ful+10; Leh11] are used for a better user experience.

A simple algorithm (none of the algorithms described later in this work) downloads or buffers annotations up to a certain point in an application cache. The advantage of this cache is that the elements are kept in the cache if the application is interrupted, for example from a phone call. A scheduler manages all the elements that are displayed with the video in order to refresh the annotations displayed, when fast-forward or fast rewind stops or the user jumps to a point in time in the scene. For a detailed description of the SIVID Player see [Kï1; MKK11].

An **iPhone app** was implemented in Objective C using the iOS SDK. It also meets the requirements described for the web player in [Mei+10a]. It reads the XML file with the description of the order of the scenes and the times when annotations are displayed and hidden. Different actions are triggered to load new scenes, to display annotations, to show a button panel for the selection of the next video, or to end the video. A video control for play, pause, and stop is shown as well as a table of contents. The GUI of this app was designed according to the results of a paper prototyping [Sny03] and a user survey [SD94]. A user evaluation was carried out on an iPhone with a running implementation of the player app at the end of the implementation phase. We used a usability walk-through [Wha+94] combined with the think aloud method [SBS94]. Results of these studies can be found in [Lan11]. The GUI of the iPhone-app is quite similar to the GUI of the Android-app. Some ways of performing a function like downloading a video to the device or invoking the search function are different due to the underlying operating system. For a detailed description of the iPhone app see [Lan11].

3.3.1. Non-linearity

Extended forms of navigation can be divided into two categories: navigation at the end of scenes and global navigation. Navigation at the end of a scene is implemented as a simple button panel. Global navigation is realized by a table of contents and a search function. The implementation of non-linearity is quite similar in desktop and both mobile players.

Button panel The next scene is triggered by a click on a button in the button panel (see Figure 3.16). The button panel is merged from the rhombus and the adjoining rectangles in the scene graph. The latter ones are linked to scenes.

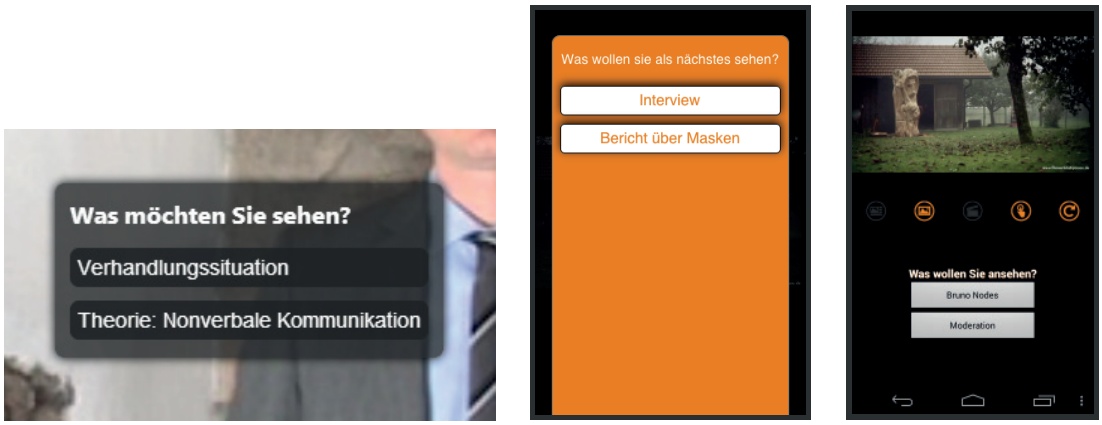


Figure 3.16.: Selection panel in the different players: HTML5 desktop player (left), iPhone mobile player (center), and Android mobile player (right).

Table of contents A table of contents (see Figure 3.17) is either shown in one of the side areas (in the HTML5 player) or can be activated by clicking on the respective button in the control bar (in the desktop and both mobile players). Entries in the table of contents are linked to scenes of the underlying scene graph. The user jumps to a point in the scene graph and can watch the video from thereon.

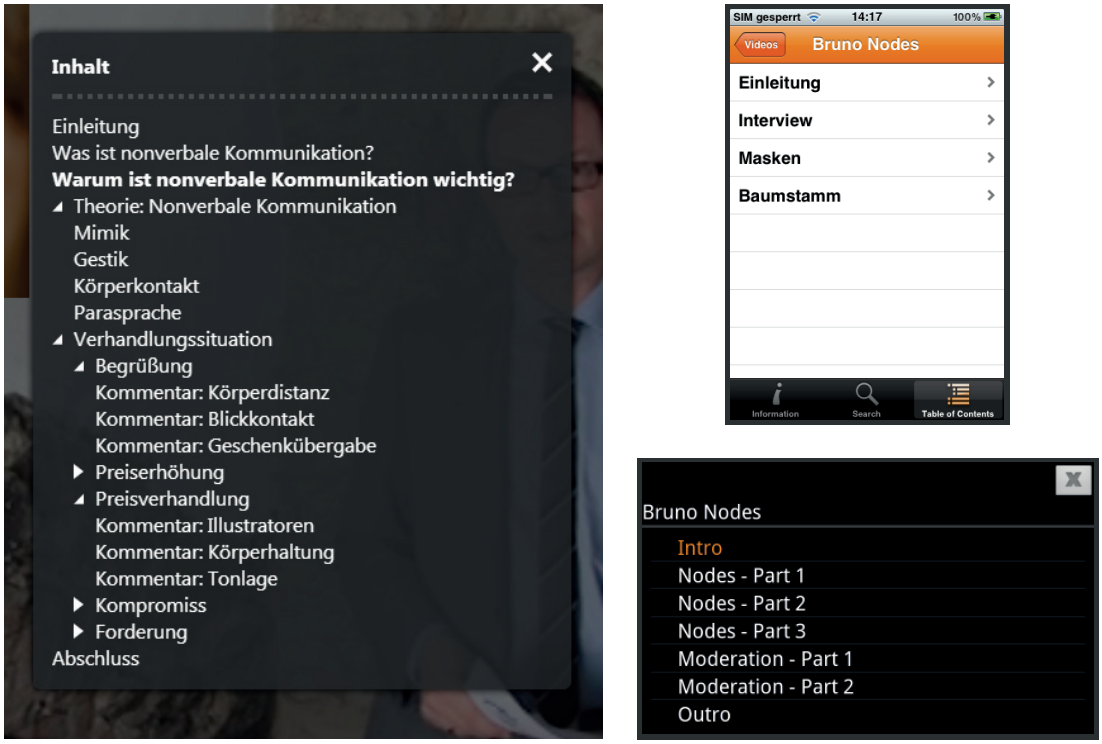


Figure 3.17.: Table of contents in the different players: HTML5 desktop player (left), iPhone mobile player (right, top), and Android mobile player (right, bottom).

Search A search function which refines the results during user input offers links based on the search result on keywords. These are either linked to a scene (see Figure 3.18) or an additional information in a scene.

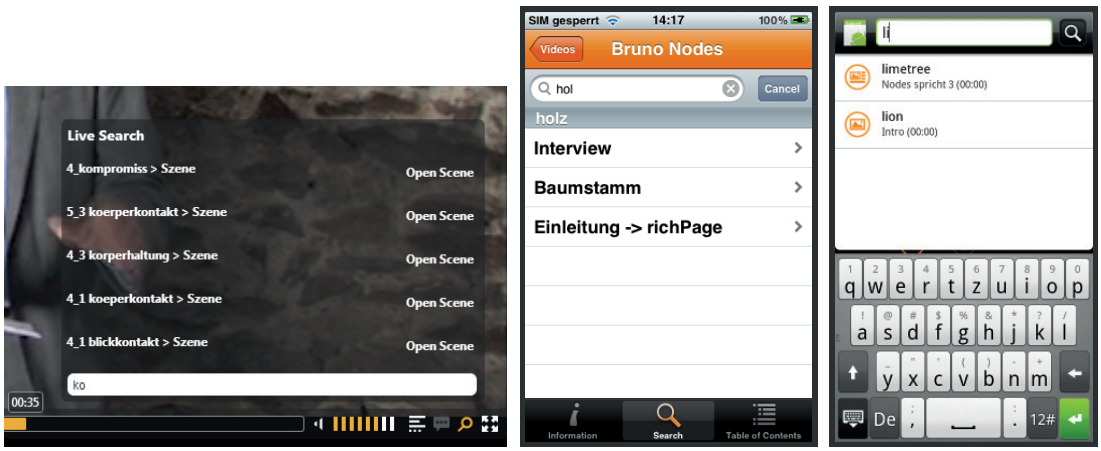


Figure 3.18.: Search function in the different players: HTML5 desktop player (left), iPhone mobile player (center), and Android mobile player (right).

3.3.2. Annotations

Additional information is displayed and hidden by the players according to the definitions in the XML file. The desktop player shows each media file in its assigned area, while the mobile players provide only one area for additional information.

Display of additional information The HTML5 player is capable of showing additional information in up to four areas grouped around the main video. Figure 3.19 (left) shows an example with two areas (right and bottom side). Additional information is shown in areas around the main video (as in Figure 3.19) or as an overlay over the video. The time when single annotations are displayed or hidden is either defined by an XML file and triggered by the player, or it is activated by a user clicking on a hotspot in the video. The progress bar indicates the display of additional information by showing thin lines at the corresponding points in time.

The mobile player has to deal with further constraints. Additional contents cannot be displayed in annotation areas arranged around the main video because of the lack of display space. An annotation observer button, at one side of the display, indicates that one or more annotations of a certain type are available at a particular point in time. The viewer can chose, if the annotation should be displayed by touching the annotation observer. The SIVID Player has three display modes therefor. The video full screen mode only shows the video and the annotation observer. A split screen mode (see Figure 3.19, right top) shows the video and a scrollable annotation stack. A long touch on the annotation pauses the main video and displays the annotations full screen (annotation full screen mode). This is very useful if the scene is annotated with long texts or annotations the viewer wants to view in detail. The display modes are implemented for two orientations, the portrait and the landscape mode. The screen modes are similar in the iPhone player (see Figure 3.19, right bottom).

Figure 3.19.: Display of annotations in the different players: HTML5 desktop player (left), iPhone mobile player (right, bottom), and Android mobile player (right, top).

3.3.3. Interactivity

Different forms of interactivity exist in our players. More interaction with the player is necessary using a mobile player. Due to the smaller display size on smart phones, less information can be displayed at a time. Consequently, the user has to initiate the display of new information which may be covered by a previously shown information. The collaboration function is not yet implemented in the mobile players, it is only described for the HTML5 player.

Built-In Player Buttons Interactive features of a standard video player like play and pause, a volume control, mute/unmute, and a progress bar are provided by each player. The extended control bar of the HTML5 player furthermore contains buttons to display a table of contents (if the table of contents is not displayed in one of the side areas), to add/show user-generated annotations, to show a search function, and to change views. The latter enables the user to switch into full screen mode, to hide all annotations, and to fade subtitles and annotation markers in or out. It is also possible to switch between two views. One shows the video and the annotations in separate areas. The other shows the video over the full width of the window and the annotations as semitransparent overlays over the video. The mobile apps also provide standard player controls. They furthermore have buttons which show an annotation on a click. These buttons are only shown, if some corresponding additional information is active at a time. The SIVID player shows how many new additional information are available for display at a time by adding small circles with numbers to the buttons. The buttons for the keyword search and the table of contents are positioned differently in both mobile players. The Android player provides a two-part button bar, because of the limited space on a smart phone. Thereby, the first part contains the standard controls and the table of contents button. The second part provides the search, the language, the bookmarks, and the save button. Both parts have a button to switch between them (left button in Figure 3.19, right, top). The iPhone player has a button to pause and leave the video and enter another screen which shows tabs with information about the video, a table of contents, and the keyword search.

Collaboration An enhancement of interactivity for the user is gained by adding collaborative capabilities to the HTML5 player [Wei12]. This enables the user to add text and images to an existing video and share this information with other users. After acquiring the log in data from a simple user management component and logging into the player, annotations can be added to the video (see Figure 3.20, left). The annotation type has to be selected first (see Figure 3.20, center), the content is added after that. Either a text is edited in a basic text editor (TinyMCE[11]) or an image is uploaded from the file system. The next step is the positioning of the annotation in either one of the side areas or as an overlay over the video area. Size and position have to be determined on a grayed out player view in the latter case (see Figure 3.20, right). Positioning of an annotation in a side area is accomplished by clicking in the desired area. The last step is the definition of the display-time of an annotation. Therefor, the markers for start- and end-point are positioned on the timeline. The annotation is saved to the video project by clicking the accept button. It is displayed to the other users when they load the scene. The author of a collaboratively added information is displayed on mouse-over. An annotation can be deleted by the owner during its display. A dust bin icon is shown on mouse over. After clicking it, the deletion has to be confirmed. Added annotations with all associated

[11] `http://www.tinymce.com/` (accessed April 26, 2014)

metadata are stored in the database. Collaborative functions in the mobile players need to be designed wisely. Paper prototyping or other usability evaluation methods may be necessary to create an intuitive GUI which can be used efficiently on the small display.

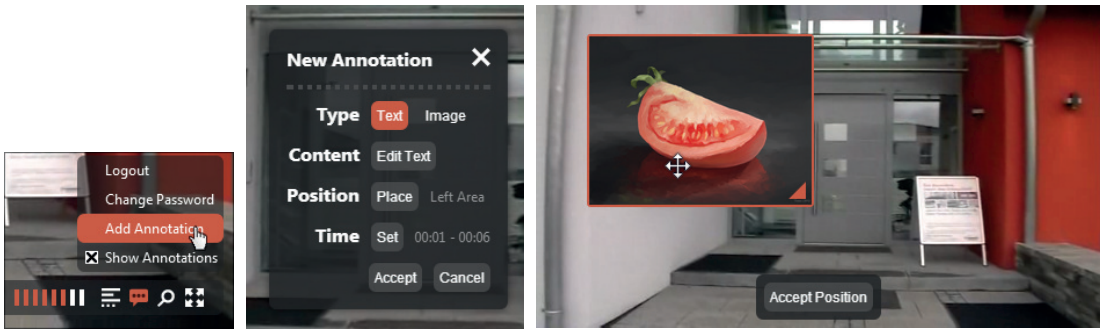


Figure 3.20.: Collaboration function in the HTML5 player: add a new annotation (left and center), position the new annotation (right).

Hotspot Hotspots in the video can be clicked by users to invoke the display of additional information provided with an object in the video. A click on an image annotation (or an image in an image gallery) loads that image as an overlay over the video for closer examination. Additional information may also be displayed in one of the annotation areas. No big differences in the appearance and behavior can be recognized between the desktop and the mobile players.

(Parts of this section (3.3 SIVA Player) were taken and adapted from our previous works [MKK11] and [Mei+13].)

3.4. Summary

In this section we described the different parts of the SIVA Suite, namely an XML format, an authoring tool, and several players. They all provide functions for interactivity, non-linearity, and to provide additional information. Innovative functions are for example clickable hotspots marking an object in the video, markers on the timeline, selection panels after scenes, a table of contents, a keyword search, and different types of additional information like rich text, images, audio files, and videos.

The software can be used to create annotated interactive non-linear videos for various scenarios. A logging component in the HTML5 player allowed us to analyze the user behavior during playback and to gather information about end user devices, bandwidth restrictions, and used browser software. The current versions of all players have no download or cache management implemented which might reduce download volume or waiting times at the beginning of scenes at forks. We conducted a user study with the HTML5 player at which users could make suggestions for improvements after watching the video a certain time. The sometimes very detailed comments in the free-text field revealed hints for usability improvements. One suggestion was to provide contents with a lower resolution to decrease loading times. Already watched scenes should be cached at client side and contents of a scene should be pre-cached. The intention of using our logging data for a goal-oriented pre-loading and caching of video parts and annotations cannot be realized in HTML5 due to the little influence given by the

`preload` and the `autobuffer` attribute. It is for example not possible to request a specified range of frames of a video from the server or to keep frames in the cache which might be needed in the future playback of the video.

Despite the fact that a download and cache management cannot be implemented with recent technologies like HTML5, we think it is important to find ways to decrease the overall download volume and waiting times at the beginning of scenes. The remainder of this work searches for algorithms and strategies which are capable of accomplishing these goals. The algorithms and strategies are developed and evaluated with a web-standard independent simulation framework. Results can than be applied to real world player implementations as far as the underlying technologies allow their realization.

4. Techniques and Methods for Download and Cache Management

A combination of algorithms and strategies is needed for downloading, playing, caching, and deleting elements. In this section, we first describe different implementations of non-linearity in players, which has influence on our algorithms and strategies. Furthermore, we analyze possible user behavior during video playback and how other authors deal with it. In addition, we study possible algorithms for the scheduling of download queues, approaches dealing with techniques for download and streaming of annotated interactive non-linear video, as well as cache management and replacement strategies. (This paragraph was taken and adapted from our previous work [MH12].)

4.1. Player Implementations for Non-linearity in Videos

An evaluation of web-players for interactive and/or non-linear video showed, that three groups of non-linearities combined with time-based annotations can be found: non-linearity by linking, non-linearity by time leaps, and "real" non-linearity. All three ways of non-linearity require some kind of clickable area to trigger or select the loading of the follow-up scene. Non-linearity by linking is the simplest method, because standard web player implementations with small extensions can be used. The next scene is selected by a hyperlink with a URL. On scene change, a new web page with an embedded video is loaded (see Figure 4.1 (c)). Non-linearity by time leaps is realized by jumps to defined frames in one linear video. The player needs a method to load a certain frame and play the video from there on. This behavior is represented in Figure 4.1 (a). "Real" non-linearity loads the new scene in the player itself as illustrated in Figure 4.1 (b). The video appears as a unified whole thereby.

Players like the Viddix player [VID10] and YouTube Video Annotations [You13] provide non-linearity by linking. This behavior destroys the fluid video experience and the video itself does not form a unified whole anymore, because they load a new web page with an embedded video and start buffering the video only then. Annotations are loaded with the video, no buffer time can be noticed when an annotation loads. Players from Quick.tv [Qui10] and VideoClix [Vid12] allow non-linearity by time leaps. The viewer is able to jump to a defined time in the video where a new scene starts or interesting contents are shown. Both players provide different buffer strategies. The VideoClix player shows the available jump labels only if the whole video is loaded. They are displayed as markers on the timeline of the video. The Quick.tv player shows the jump labels as illustrated links overlaid over the video but provides no buffering function. So the viewer has to wait until enough frames are loaded to play the video. Text and small image annotations are loaded after user interaction. Therefore, no or only a short buffer time can be observed. "Real" non-linearity is implemented in the XIMPEL player [Bhi+10]. The viewer is asked what he wants to do at the end of a scene. Then he can choose how the video should go on from there. The next scene starts playing after a short

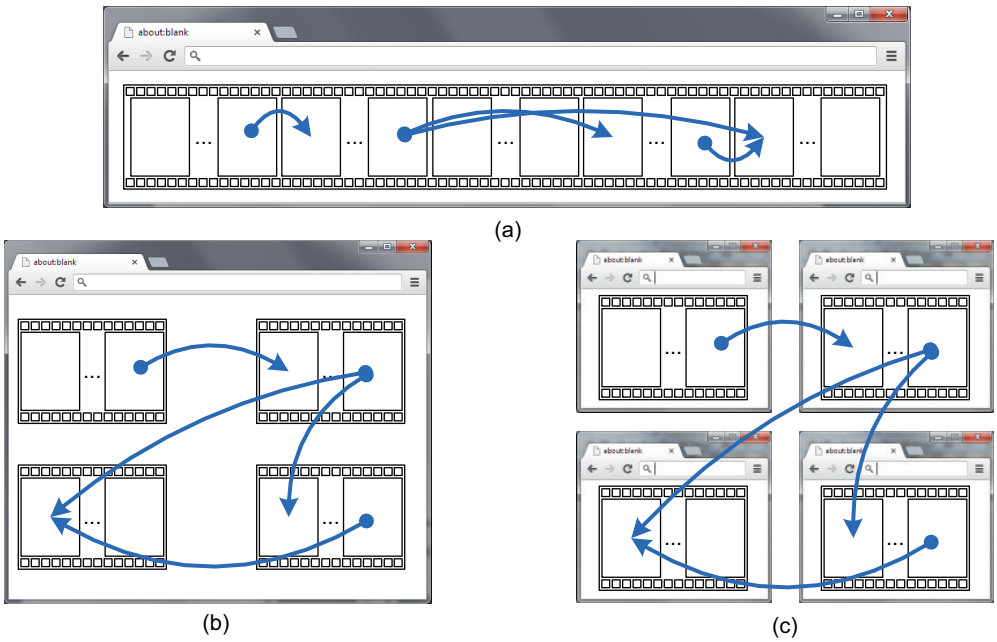


Figure 4.1.: Different ways of implementing non-linearity in videos: (a) non-linearity by time leaps, (b) "real" non-linearity, (c) non-linearity by linking.

loading time. No buffer time can be noticed when annotations (mainly text or small images) load. A more detailed overview of these and other players can be found in Section 2.4.

Critical reflection: Only the implementation of "real" non-linearity is suitable for annotated interactive non-linear videos as defined in this thesis. The impression of one single video increases the viewing experience. Furthermore, preloading of future scenes can be realized better with "real" non-linearity, because smaller video scenes are easier to handle for download than one linear video from which different parts need to be preloaded. "Real" non-linearity and non-linearity by linking allow the definition of graph structures without limitations. Non-linearity by time leaps does not allow the playback of scenes in a different order because jumps on the timeline have to be triggered by the user. Furthermore, it is not possible to link different annotations to a scene depending on the course of the video, because annotations are linked with video time. A table of contents and an index or keyword search can be implemented with each form of non-linearity. Non-linearity by linking requires the integration of these elements in each website which is part of the non-linear video. A summary of

	Viewing experience		Non-linear structures		
	Impression of one video	Preloading	Graph	Table of contents	Index
Non-linearity by linking	−−	−−	+	$+^1$	$+^1$
Non-linearity by time leaps	+	0	−−	++	++
"Real" non-linearity	++	++	++	++	++

Table 4.1.: Suitability of forms of non-linearity for annotated interactive non-linear videos ($+^1$: these functions require separate structures).

these analysis is presented in Table 4.1, whereby "++" indicated a high suitability, "0" represents neutrality, and "−−" shows unfitness. (This section (4.1 Player Implementations for Non-linearity in Videos) was taken and adapted from our previous work [MH12].)

4.2. User Behavior during Video Playback

User interaction in traditional linear videos is generally limited to VCR actions like play, pause, stop, fast-forward, and fast rewind, as well as jumps on the timeline (see gray area in Figure 4.2). The integration of interactivity and/or non-linearity, like a table of contents, a keyword search, selection panels, quizzes, or hotspots to select a consecutive scene, adds navigational functions to a player. Furthermore, interaction with the video in form of tilt, zoom, or panoramic navigation may be implemented in a player.

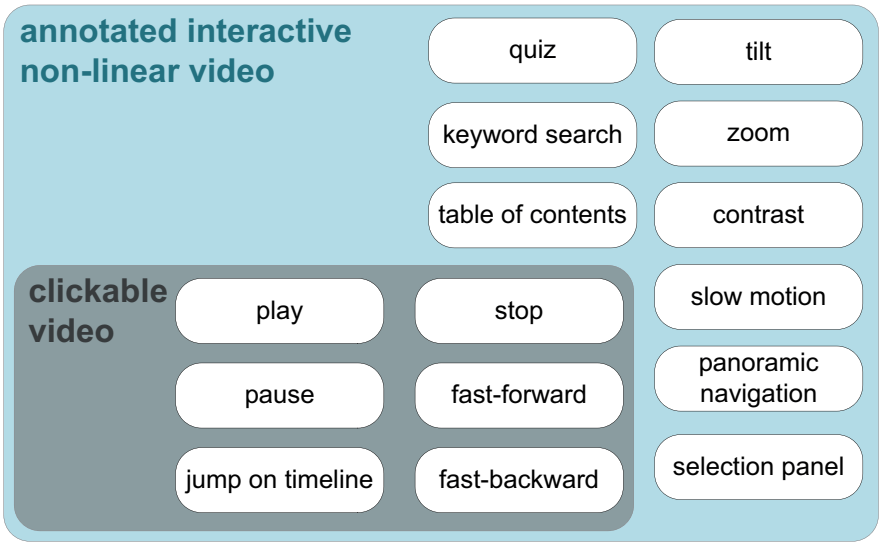


Figure 4.2.: Functions of traditional linear videos and additional functions of annotated interactive non-linear videos.

We first want to discuss two works in more detail, because they are very closely related to our approach. The first one deals with interactivity in linear videos, the second one with a latency reduced streaming of hypervideo.

- **Interactivity in linear videos**
  Interactivity in linear videos is addressed by Fei et al. in [Fei+99; Fei+05]. They "propose an active buffer management technique to provide interactive functions in broadcast VoD systems" [Fei+05]. The buffer management in broadcast VoD systems is the major part of their work, but they also introduce a user interaction model for VCR-actions. They furthermore add probabilities for performing a VCR action and duration ratios for the VCR actions to the model. These are used to model certain user behaviors while watching a video. Using that model, they deal with active buffer management for video on demand systems. A kind of sliding window is proposed to make past, present, and future parts of the video available.

Critical reflection: This work deals with VCR functions in linear video sequences sent over multicast frameworks. It uses client side buffering to allow the user to perform VCR actions. Basic parts of this work can be implemented in our framework, but they have to be extended or altered before being able to deal with annotations and non-linearity. The principle of the sliding window may be used in our algorithms/strategies for cache and delete management. The user interaction model can furthermore be extended to simulate interactions in annotated interactive non-linear videos.

- **Latency reduced streaming in hypervideos**
  In the work of Grigoras, Charvillat, and Douze [GCD02], hypervideo is streamed, taking user behavior into account. Their "objective is to optimize hypervideo prefetching [sic] in order to reduce the latency caused by the network" [GCD02]. More precisely, they want to prevent bandwidth under-usage in case of streams with a "bitrate lower than the bandwidth" and provide a fluent stream when the "bitrate is greater than the bandwidth" by pre-fetching data. The authors distinguish between long-term and short-term memory. Long-term memory is "based on a (cumulative) statistical analysis of interactions of users from various communities or sessions" and short-term memory "refers to the most recent interactions of one or multiple users" [GCD02]. Using that, the authors want to predict which contents are needed for a fluid playback. They outline, that they need a model which describes possible user interactions and algorithms for prediction and pre-fetching of the contents. The authors describe a simple model with two pre-fetching policies called proportional and best-first policy. The proportional policy works in a way that "all states that can be reached are prefetched [sic] from the current state. The bandwidth allocated for each stream is proportional to the probability of the corresponding transition" [GCD02]. The best-first policy pre-fetches only the most probable state. Each policy has two versions, a conservative and an aggressive. The conservative version "tries to use as little bandwidth as possible, and stops prefetching [sic] as soon as the amount b (minimum size needed to begin playing) of the stream has been downloaded" [GCD02]. The aggressive version "continues downloading and tries to use all available bandwidth" [GCD02]. This model only uses long-term memory and applies only one fixed probability for a transition in a hypervideo at playback. This leads to the problem, that, if a user visits a scene a second time, the same probability is assigned as for visiting it for the first time, but the authors assume that the probabilities should be different at a second visit due to a realistic user behavior. Solving this problem, the authors use short-term memory as well as buffer states and formulate a formal Markov Decision Problem. An optimum policy is predicted. They optimize the Markov Decision Process with stochastic dynamic programming and value iteration. According to that implementation of the policy, buffer management is implemented and the player is started.

  Critical reflection: This work integrates intelligence into the decision process at forks in the video flow. This approach could be used in our approach to provide the probabilities for downloading a scene at a fork automatically. However, the authors do not describe how to solve the cold start problem. They assume that each hypervideo has a viewing history. This work only deals with the decisions at forks in the video flow, neither annotations are added to the video which add additional download volume, nor do the authors describe any cache or delete management.

Other works from this area are not as closely related to our work as these proposed by Fei et al. and Grigoras, Charvillat, and Douze. Kozuch, Wolf, and Wolfe define different client models for video libraries in [KWW00]. "The FastDVL model represents a moderately non-linear usage pattern, such as might be the case when a number of clients are displaying video presentations. [...] The SlowDVL model represents a lesser degree of nonlinearity [sic]. This pattern of usage might arise in situations where most of the users are navigating/browsing through video titles. Such users might play clips of interest for some time, while disregarding others after a short play time. [...] The VoD model represents user accesses with nearly zero non-linear access" [KWW00]. This is one of the few works taking non-linearity into account, but deals with a not further specified and more unstructured form on non-linear videos than our work does. Their videos range from simple video playback to playback of videos from a video library which form a strongly connected graph.

Laraspata, Striccoli, and Camarda [LSC10] describe a scheduling algorithm for interactive video called SAIV for variable bit rate (VBR) video stream transmission in UMTS networks "varying the sampling frequency of the Real Time Control Protocol (RTCP)" [LSC10]. User interactivity may be pause, fast-forward, fast rewind, and so on. No further forms of interaction as provided by our annotated interactive non-linear videos are described. The proposed algorithm could be used to adapt the videos in networks with smaller bandwidths. Makar et al. [Mak+10], Inoue et al. [Ino+10], Halawa et al. [Hal+11], and Khiem, Ravindra, and Ooi [KRO11] describe algorithms and user studies for streaming videos where the user can interact with the video in different ways. Tilt, zoom, and panoramic navigation are functions to interact with the video in order to get more detailed or varying information.

Hollfelder, Friedrich, and Aberer describe two different kinds of prediction logic for "varying consumption rates due to users' interactive behavior" in [HA98] and [FHA00]. They compared both of them in [HA99]. In one approach, the future consumption is predicted from the system behavior in the past. The other approach deduces a prediction from a user behavior model. Their works focus on sessions of clients on multimedia servers and an admission control if enough resources are available. In [FHA00] the grant of admission is based on a Continuous Time Markov Chain model (CMTC) which can calculate possible starting points for the download management of annotated interactive non-linear video, but is applicable only server-sided. In [HA98] a framework for admission control to the multimedia server and scheduling of requests of clients which were already admitted to the server is introduced. This work is server-based, too. (This section (4.2 User Behavior during Video Playback) was taken and adapted from our previous work [MH12].)

Critical reflection: Related work taking user behavior into account considers VCR functions, non-linearity, or interactivity. Neither of the works deals with a combination of these characteristics as enlisted in Table 4.2. The focus of these works is either on VCR actions or on certain interactive or non-linear features, but not on a combination of them. VCR functions in combination with non-linearity are important for annotated interactive non-linear videos due to their structure and the navigation behavior of viewers. Two important (and most related) works from these areas are those from Fei et al. [Fei+05] and Grigoras, Charvillat, and Douze [GCD02]. Fei et al. [Fei+05] deal with client side buffering using a sliding window to make a certain amount of frames available from a play point forwards and backwards. This mechanism is needed in annotated interactive non-linear videos within a scene, for a history of already watched scenes, and for all possible future scenes at a fork. In the latter case, the approach described by Grigoras, Charvillat, and Douze [GCD02] can be used to pre-fetch elements at forks in the video flow.

	VCR functions								Non-linearity and interactivity					
	Play backward	Slow forward	Slow rewind	Pause	Stop	Fast-forward	Fast rewind	Jump on timeline	TOC/Index	Scene selection	Pan/panorama	Tilt	Zoom	Contrast&navigation
Fei et al. [Fei+99; Fei+05]	✓	✓	✓	✓	–	✓	✓	✓	–	–	–	–	–	–
Friedrich et al. [FHA00]	–	–	–	✓	–	✓	✓	–	–	–	–	–	–	–
Hollfelder & Aberer [HA98]	–	–	–	✓	–	✓	✓	–	–	–	–	–	–	–
Laraspata et al. [LSC10]	–	✓	–	✓	✓	✓	–	–	–	–	–	–	–	–
Grigoras et al. [GCD02]	–	–	–	✓	✓	–	–	–	–	✓	✓	–	✓	✓
Halawa et al. [Hal+11]	–	–	–	–	–	–	–	–	–	–	✓	–	✓	–
Inoue et al. [Ino+10]	–	–	–	–	–	–	–	–	–	–	✓	–	–	–
Khiem et al. [KRO11]	–	–	–	–	–	–	–	–	–	–	✓	–	✓	–
Makar et al. [Mak+10]	–	–	–	–	–	–	–	–	–	–	✓	✓	✓	–
Kozuch et al. [KWW00]	–	–	–	–	–	–	–	–	✓	–	–	–	–	–

Table 4.2.: Consideration of user behavior in related work about linear and interactive non-linear videos with focus on VCR functions, non-linearity, and interactivity.

4.3. Scheduling of Download Queues

All elements of an annotated interactive non-linear video, namely frames and annotations need to be transmitted from the server to the client. This can be accomplished in a serial or a parallel way with varying order of the single elements, but it is important to take the weights which are applied to single paths into account. The used scheduling algorithm should provide the following characteristics:

- It should be able to assign weights to queues.
- The order inside a queue needs to be kept.
- A serialization of the elements into one single download queue is preferred to multiple queues.
- Incoming flows should not be rearranged.
- The algorithm should be fair and make propositions regarding to possible delay.

Only those scheduling algorithms which provide these features are studied more closely in this section. Scheduling problems appear in many different areas of research and therefore there exists a large number of different algorithms dealing with varying requirements. Surveys on algorithms from the areas of operational research, constraint programming, and more general algorithms can be found in Allahverdi et al. [All+08], Bartak, Salido, and Rossi [BSR10], Potts and Strusevich [PS09], Lombardi and Milano [LM12], and Shabtay, Gaspar, and Kaspi [SGK12]. The algorithms collected in these works are too complex with regard to our require-

ments because they are taking too many side conditions into account. Furthermore, mainly multiple queues are getting served by the algorithms while we only have one queue.

Surveys on scheduling in P2P networks, on real-time multiprocessor scheduling, and on pre-emptive/non-pre-emptive scheduling for processors written by Yue et al. [Yue+11], Davis and Burns [DB11], and Buttazzo et al. [BBY13] deal with more complex problems and underlying structures than we have in our scheduling task. Davis and Burns differentiate three areas of attributes, namely allocation, priority, and interruptibility. Neither allocation nor interruptibility are interesting for our work. We do not analyze work from these areas in more detail therefore.

Lin, Hong, and Lin investigate a "sequence optimization of media objects in a multimedia presentation that is dynamically composed from digital libraries" [LHL13]. They "formulate the sequencing problem with buffer constraints in the media player into a flowshop scheduling problem and present a reduction strategy with a branch and bound algorithm to derive optimal sequences" [LHL13]. Their algorithms take "buffer constraints" and "due-date constraints" into account. They integrate their strategies and algorithms into a Flash-based prototype system implementation consisting of a multimedia database, a bandwidth estimation module, a scheduling engine, an object pre-fetcher, and a media presenter. Their results show that their algorithms work better than earliest-due-date-based sequences of multimedia objects. Some aspects of this work are interesting for annotated interactive non-linear videos, but there are major differences: The contents have no fixed structure, they vary from query to query. Furthermore, no assumptions can be made according to the need of future elements. The delete scheduling is not taken into account.

Scheduling algorithms from the network area are Weighted Fair Matching (WFM) described by Lee, Shin, and Youn [LSY04], Deficit Round Robin with Fragmentation (DRRF) outlined by So-In, Jain, and Tamimi [SIJT10], as well as the approaches described by Jalali, Padovani, and Pankaj [JPP00], Song, Lin, and Cruz [SLC08], and Vasiliadis, Rizos, and Vassilakis [VRV12]. These algorithms might be applicable to our problem with some adaptations, but provide much more features and more complex underlying structures than needed for the solution of our scheduling problem. The same applies to algorithms for single- or multiprocessor scheduling like Distributed Weighted Round-Robin (DWRR) presented by Li, Baumberger, and Hahn [LBH09], Fair-Priority-Expression-Based (FairPEB) burst scheduling described by Shi et al. [Shi+09], Earliest Deadline First (EDF) outlined by Sohn and Kim [SK97], or the approach described by Zotkin and Keleher [ZK99].

Generalized Processor Sharing (GPS) is described by Parekh and Gallager as a "flow-based multiplexing discipline" which is "efficient, flexible, and analyzable" [PG93]. "[...] When combined with Leaky Bucket admission control, [it] allows the network to make a wide range of worst-case performance guarantees on throughput and delay" [PG93]. A GPS server "is work conserving and operates at a fixed rate r" [PG93]. Throughput guarantees can be made if the average rate of a session is smaller than the guaranteed rate of a session. "The delay of an arriving session i bit can be bounded as a function of the session i queue length, independent of the queues and arrivals of the other sessions" [PG93]. Sessions can be weighted. Furthermore, "it is possible to make worst-case network queueing delay guarantees when the sources are constrained by leaky buckets" [PG93]. GPS has similar requirements compared to our scheduling problem, while proposed solutions from this area have similar underlying structures. For that reason, some important algorithms from this area of work are described hereafter in more detail:

- **FCFS**: First Come First Served uses a single queue in which new elements are inserted at the end [DSS94].
- **FQ**: Fair Queuing is capable of "fair allocation of bandwidth, lower delay for sources using less than their full share of bandwidth, and protection from ill-behaved sources" [DKS89]
- **WFQ/PGPS**: Weighted Fair Queuing is capable of applying weights to single input queues. Parekh and Gallager rename it to PGPS and describe it as "a practical packet-by-packet service discipline, [...], that closely approximates GPS" [PG93].
- **WF^{2}Q**: Worst-case Fair Weighted Fair Queuing (WF^{2}Q) is "a new packet approximation algorithm of GPS" [BZ96b]. The provided service "is almost identical to that of GPS, differing by no more than one maximum size packet" [BZ96b].
- **WF^{2}Q+**: Another Worst-case Fair Weighted Fair Queuing called WF^{2}Q+ is proposed by Bennett and Zhang in [BZ96a]. It provides "the tightest delay bound among all PFQ algorithms" and has "the smallest WFI among all PFQ algorithms" [BZ96a].
- **WF^{2}Q-M**: A Worst-case Fair Weighted Fair Queuing scheduling scheme which supports "maximum rate control and minimum service rate guarantee" called WF^{2}Q-M is proposed by Lee, Sun, and Chen in [LSC03]. It "proposes the virtual clock adjustment method to enforce maximum rate control by distributing the excess bandwidths of maximum rate constrained sessions to other sessions without recalculating the virtual starting and finishing times of regular sessions" [LSC03].
- **DRR**: An approximation of fair queuing called "Deficit Round Robin" is described by Shreedhar and Varghese in [SV95]. This algorithm "achieves nearly perfect fairness in terms of throughput and is simple enough to implement in hardware" [SV95] according to the authors.
- **PWFQ**: Priority-based Weighted Fair Queuing "combines a session's allocated share to achieve the bandwidth guarantee and the session's priority to adjust the delay bound inside a sliding window" [WWL02]. It furthermore "decouples the delay from the service share so that a session with a low share but a high priority may still receive a small delay" [WWL02].
- **BSFQ**: Bin Sort Fair Queueing combines the strengths of sorted priority methods (excellent approximation for WFQ) and frame-based methods (computationally efficient) [CP02]. "BSFQ is highly scalable [...] [and] can provide end-to-end delay and fairness guarantees to conformant flow" [CP02].
- **WBSQ**: Worst-case Fair Bin Sort Queuing provides "good worst-case fairness and delay properties, yet has low complexity and is amenable to simple hardware implementation" [DR11]. Therefor, it combines "features of BSFQ and WF^{2}Q+" [DR11].
- **SCFQ**: Self Clocked Fair Queuing is "based on the adoption of an internally generated virtual time as the index of work progress" [Gol94]. According to the authors, it "is nearly optimal, in the sense that the maximum permissible disparity among the normalized services offered to the backlogged sessions is newer more than two times the corresponding figure in any packet-based queueing system" [Gol94].

- **FFQ**: Frame-based Fair Queuing "provide[s] the same bounds on end-to-end delay and buffer requirements as those of WFQ" [SV98]. It "uses a framing mechanism to periodically recalibrate a global variable tracking the progress of work in the system, limiting any short-term unfairness to within a frame period" [SV98].
- **SPFQ**: Starting Potential-based Fair Queueing "provide[s] the same bounds on end-to-end delay and buffer requirements as those of WFQ" [SV98]. It "performs the recalibration at packet boundaries, resulting in improved fairness" [SV98].
- **SWFQ**: Simple Weighted Fair Queuing is an "effective scheduling algorithm based on the RPS model" [Wan+01]. It has a comparably low complexity and does not "require such division or multiplication operations" [Wan+01] as other algorithms.
- **SFQ**: Start-time Fair Queuing is "computationally efficient, achieves fairness regardless of variation in a server capacity, and has the smallest fairness measure among all known fair scheduling algorithms [in 1996]" [GVC96]. The authors claim that this algorithm is "better suited than Weighted Fair Queuing for integrated services networks and it is strictly better than Self Clocked Fair Queuing" [GVC96].
- **Delay-EDD**: Delay-EDD (earliest due date) is the attempt to provide "real-time services on a packet-switched store-and-forward wide-area network with general topology" [FV90]. Thereby, channels are established "with deterministic or statistical delay bounds" [FV90].
- **VirtualClock**: The "VirtualClock maintains the statistical multiplexing flexibility of packet switching while ensuring each data flow its reserved average throughput rate at the same time" [Zha90].

	Weighting	Fairness	Delay guarantees
FCFS [DSS94]	–	✓	–
DRR [SV95]	✓	✓	✓
WFQ/PGPS [PG93]	✓	✓	✓
EDF [SK97]	–	–	✓
FQ [DKS89]	–	✓	–
PWFQ [WWL02]	✓	✓	✓
WBSQ [DR11]	–	✓	✓
SCFQ [Gol94]	–	✓	✓
Delay-EDD [FV90]	–	–	–
Virtual Clock [Zha90]	–	–	✓
FFQ [SV98]	–	✓	✓
SPFQ [SV98]	–	✓	✓
SFQ [GVC96]	–	✓	✓
WF^2Q[BZ96b]	✓	✓	✓
WF^2Q+[BZ96a]	✓	✓	✓
WF^2Q-M [LSC03]	✓	✓	✓
BSFQ [CP02]	✓	✓	✓
SWFQ [Wan+01]	✓	✓	✓

Table 4.3.: Characteristics of different scheduling algorithms in the domain of Generalized Processor Sharing (GPS).

Critical reflection: Except for Delay-EDD and VirtualClock, all scheduling algorithms from the list in this section are applicable for our problem with some adaptations. We prefer an algorithm with a low complexity, because in case of changes in the video flow, existing download queues are discarded and new ones are created, which should be done in an efficient way. Furthermore, the weighting of queues and not of single elements in a queue is necessary, because our queues are sorted before they are handled by the algorithm. For that reason, a rearrangement of queues is prohibited. Delay guarantees for certain elements in the queue should be given although the effect of one delayed element may be small. Table 4.3 gives an overview of the already described algorithms. We take these into consideration which provide weighting, fairness, and delay guarantees. From the remaining three algorithms we chose DRR because of its very good applicability to our scheduling problem with small adaptations.

4.4. Download and Streaming of Interactive (Non-linear) Video

Great strides have been made in the past 15 years in the area of transporting videos from (web) servers to clients. In the beginning, the only concern was to deliver the data in an efficient way to reduce access latency, bandwidth, and storage usage at the client side. Nowadays more sophisticated streaming algorithms and frameworks for multimedia data over different network-types exist. An error correction mechanism for video streaming over wireless networks is proposed by Tsai et al. [Tsa+10]. Wireless multimedia delivery over 802.11e with cross-layer optimization techniques is described by Chilamkurti et al. [Chi+10]. Lee and Park [LP10] suggest a scalable and adaptive video streaming framework over multiple paths. A scalable multimedia QoS architecture for ad hoc networks is described by Mehmood and Alturki [MA11]. A content based delivery network-based streaming architecture for wireless IPTV is described by Palau et al. [Pal+11].

Further developed cache management and streaming techniques for linear video are described in Lee and Chung [LC08], Liebl et al. [Lie+05], Sharman et al. [Sha+07], and Sun and Weng [SW12]. These approaches only address the cache needed to provide a constant stream for video playback of linear video. No further caches to save content for later reuse are introduced. Algorithms for interactive video streaming like those described by Paluska and Pham [PP10], Xiu, Cheung, and Liang [XCL11], Fortuna et al. [For+10], Fernandez et al. [Fer+10], and Bömcke and De Vleeschouwer [BDV09] may be used for the streaming of the video contents. They may particularly be used to improve delays and waiting times while streaming or downloading the videos as an extension of our proposed algorithms. In Kosch et al. [Kos+04], heuristics are found to create an optimal or near optimal schedule for multi-clip queries. The work deals with different video streams but can be used for other kinds of files with minor changes. It describes the scheduling but not the processing or buffering of the content. Carlsson, Mahanti, and Eager [Car+08], Gotz [Got06], Mayer-Patel and Gotz [MPG07], and Zhao, Eager, and Vernon [ZEV07] describe the streaming of non-linear video or media, but do not address the cache management at the client side. In Zhao, Eager, and Vernon [ZEV07] several models for the streaming of non-linear videos are introduced and evaluated according to server bandwidth and client data overhead. CSA is proposed in [Got06] and [MPG07]. It is a framework for scalable and adaptive streaming of non-linear media to large user groups. It consists of a simple server. All work is done by the client. A decoding approach for strongly resource-restricted architectures like mobile devices is proposed by Seitner et al. [Sei+11] for linear videos. This approach can be used as a base technology for our framework.

Works from the area of TV dealing with data transmission for interactive contents like those described by Lanceria et al. [Lan+04] and Haskin and Stein [HS95] show only little commonalities with our work due to the different transmission technologies and are not further considered for this reason.

Video on demand services are usually streamed over the Internet or a broadband cable network. Depending on the used variant, different ways of interaction may be allowed. Interactive video on demand allows navigation in a linear video and shows similarities to the intra scene behavior of annotated interactive non-linear videos. Wong et al. remark, that "current state-of-the-art multicast streaming algorithms, while extremely efficient, all suffer from significant performance degradations when interactive playback controls are supported" [Won+07]. They describe and evaluate a "static full stream scheduling (SFSS)" algorithm which may improve existing multicast streaming algorithms [Won+07]. Furthermore, the approach described by Sarhan, Alsmirat, and Al-Hadrusi [SAAH10] tries to predict waiting times to give users feedback on how long they have to wait until the requested video starts. Findings from these works are not applicable to annotated interactive non-linear videos because we do not stream our videos. Ghose and Kim propose a survey on "Scheduling Video Streams in Video-on-Demand Systems" [GK00]. They provide "detailed discussion on policies based on principles of broadcasting, batching, caching, and piggybacking or merging. Policies like look-ahead scheduling schemes that are designed exclusively to provide certain interactive VCR-like control operations are also covered. [...] Performance of these policies in terms of bandwidth demand reduction, customer waiting time reduction, provision of interactive control by the user, and fairness of service are given special emphasis" [GK00]. They give a good overview of this area of research but provide no in-depth analysis according to the suitability of the referenced work for annotated interactive non-linear videos or hypervideos. Some aspects of their work may be considerable for our work with adaptations.

Algorithms from the area of multiview video show similarities to annotated interactive non-linear videos regarding enhanced download volumes. Kurutepe, Civanlar, and Tekalp describe a "view selective streaming technology" where only these views of the video which are displayed to the viewer are delivered from the server [KCT07]. An observer which communicates with the streaming server is used by Cheung, Ortega, and Cheung [COC11]. Based on the observations, the composition of frames is optimized due to transmission rates and storage capacities. The issues with delay in interactive multiview videos are addressed by Chen et al. [Che+09b]. They "propose a novel guaranteed service for interactive multiview video". Liu et al. [Liu+10] present "a rate-distortion (RD) optimized interactive streaming method for multiview video pre-compressed by H.264 Joint Multiview Video Model (JMVM)" which achieves a performance improvement for example compared to scalable multiview coding. The approaches described for multiview video assume that the viewer switches between views. These are different problems to deal with compared to annotated interactive non-linear videos, where no switching between video and annotations happens and all information is displayed at a time.

Though many efforts have been made in analyzing, creating, and enhancing hypervideo (see Hoffmann, Kochems, and Herczeg [HKH08], Doherty et al. [Doh+03], Shipman, Girgensohn, and Wilcox [SGW03b; SGW08], Aubert et al. [Aub+08], Aubert and Prie [AP05]), only little effort was made to optimize the data transmission from client to server or to implement a cache management at the client side in order to avoid retransmission of elements. Bota, Corno, and Farinetti [BCF02] undertook efforts to minimize the data volume transferred from server to client in transmitting hypervideo. They implemented a more efficient method

to mark hot spots in hypervideo than defining the hot spot at each frame and transferring the marks with the video. The work addresses the download of control data but not the pre-fetching of scenes or strategies for cache management. An end-to-end framework for multiple-perspective hypervideo on mobile platforms is proposed by Miller et al. [Mil+11] as work in progress. The framework consists of three components for production, delivery, and consumption of the contents. No hints on an download and cache management are given in this work which is very similar to our overall framework. Hyper-Hitchcock, a framework for authoring, viewing, and generating hypervideo is described by Shipman, Girgensohn, and Wilcox[SGW08]. The Player is evaluated for usability, but as far as the authors know, no download or cache management is implemented. Because of the similarity of the structure of detail-on-demand hypervideo and annotated interactive non-linear video, our download and cache management could be implemented in the Hyper-Hitchcock player.

Annotated interactive non-linear videos described by SMIL and played by suitable players also provide "real" non-linearity. SMIL offers functions to preload elements by using a `<prefetch>` element [BR08]. Doing this, the waiting time between scenes can be avoided (if possible). Because of the big range of functions provided by SMIL, annotations with huge amount of download time (like other videos) can be added to a video. This principle has a weak spot: The author has to specify `prefetch` elements to avoid inconsistencies in the video flow for each video in a static way. As a result, SMIL cannot adapt to former user behavior. It should be added that the `prefetch` element was not supported by all players [BR08, p. 499]. Other players like the AMBULANT SMIL player are no web-players but platform independent. It is described by Bulterman et al. in [Bul+04], but no details on download and cache management in addition to the `prefetch` element are characterized.

Gao et al. [Gao+11] describe a scheme for accurate and low-delay seeking within and across video mash-ups created with SMIL at client side. Two pre-fetching approaches are described to improve the implementation of the `prefetch` element of SMIL. These overcome the weakness of the `prefetch` element as described above. One of the approaches does pre-fetching without low-level discarding, which is more basic than the algorithm for pre-fetching with low-level discarding. The second one discards unnecessary frames immediately after decoding and not before displaying as the first one does. These algorithms can be used to provide a seamless transition between two successive scenes or a jump into a scene in order to provide a better viewing experience. (This section (4.4 Download and Streaming of Interactive (Non-linear) Video) was taken and adapted from our previous work [MH12].)

Critical reflection: The algorithms described in this section are from different areas of research, but they mainly focus on just one part of the process of transmitting the data from server to client (see Table 4.4). Some of them consider the streaming of videos, either with focus on network issues, or on cache issues. Others deal with the scheduling of data or they define how elements should be pre-fetched. Works with focus on mobile platforms or multiview videos provide some approaches for the problems we are dealing with. Furthermore, some papers focus on the quality of service or the quality of experience.

	Streaming (network)	Streaming (cache)	Streaming (to user groups)	Scheduling of data	Pre-fetch	Non-linear video/media	Hypervideo	QoS/QoE	Mobile platform	Multiview video
Tsai et al. [Tsa+10]	✓	–	–	–	–	–	–	–	–	–
Chilamkurti et al. [Chi+10]	✓	–	–	–	–	–	–	–	–	–
Lee and Park [LP10]	✓	–	–	–	–	–	–	–	–	–
Mehmood and Alturki [MA11]	✓	–	–	–	–	–	–	–	–	–
Palau et al. [Pal+11].	✓	–	–	–	–	–	–	–	–	–
Lee and Chung [LC08]	–	✓	–	–	–	–	–	–	–	–
Liebl et al. [Lie+05]	–	✓	–	–	–	–	–	–	–	–
Sharman et al. [Sha+07]	–	✓	–	–	–	–	–	–	–	–
Sun and Weng [SW12]	–	✓	–	–	–	–	–	–	–	–
Paluska and Pham [PP10]	✓	–	–	–	–	–	–	–	–	–
Xiu et al. [XCL11]	–	–	–	–	–	–	–	✓	–	–
Fortuna et al. [For+10]	–	–	–	–	–	–	–	✓	–	–
Fernandez et al. [Fer+10]	–	–	–	–	–	–	–	✓	–	–
Bömcke and De Vleeschouwer [BDV09]	–	–	–	–	–	–	–	✓	–	–
Kosch et al. [Kos+04]	–	–	–	✓	–	–	–	–	–	–
Carlsson et al. [Car+08]	–	–	–	–	–	✓	–	–	–	–
Gotz [Got06]	–	–	✓	–	–	✓	–	–	–	–
Mayer-Patel and Gotz [MPG07]	–	–	✓	–	–	✓	–	–	–	–
Zhao et al. [ZEV07]	–	–	–	–	–	✓	–	–	–	–
Seitner et al. [Sei+11]	–	–	–	–	–	–	–	–	✓	–
Wong et al. [Won+07]	–	–	–	✓	–	–	–	–	–	–
Sarhan et al. [SAAH10]	–	–	–	–	–	–	–	✓	–	–
Kurutepe et al. [KCT07]	–	–	–	–	–	–	–	–	–	✓
Cheung et al. [COC11]	–	–	–	–	–	–	–	–	–	✓
Chen et al. [Che+09b]	–	–	–	–	–	–	–	✓	–	✓
Liu et al. [Liu+10]	–	–	–	–	–	–	–	–	–	✓
Bota et al. [BCF02]	–	–	–	–	–	–	✓	–	–	–
Miller et al. [Mil+11]	–	–	–	–	–	–	✓	–	–	–
Bulterman and Rutledge [BR08]	–	–	–	–	✓	–	–	–	–	–
Bulterman et al. [Bul+04]	–	–	–	–	✓	–	–	–	–	–
Gao et al. [Gao+11]	–	–	–	–	✓	–	–	–	–	–

Table 4.4.: Focus and considered features of related work on streaming and download of (non-linear) videos.

4.5. Cache Management and Replacement Strategies

The delete process of cached annotated interactive non-linear videos and web cache replacement strategies shows similarities. As a result, the latter were evaluated for their applicability for cache clearance in our annotated interactive non-linear video player framework. According to Wong [Won06] (similar in Podliping and Böszörmenyi [PB03])[1], cache replacement policies can be sorted into five categories, namely recency-based, frequency-based, size-based, function-based, and randomized replacement policies. Not all of the categories are suitable for annotated interactive non-linear videos because of the available structural knowledge and the processing by the player. Recency-based strategies do not fit, because they are built on the design rationale that objects which have been accessed a short time ago, will be accessed again in the near future. The behavior in annotated interactive non-linear videos is opposed because the relative frequency of watching a scene which has been watched recently is usually quite small. Frequency-based strategies do not fit, too, because scenes are usually watched only once or twice in an annotated interactive non-linear video. Only on rare occasions scenes will be watched more than twice. According to this, there are no "popular" elements that have to be kept in the cache. The last category of strategies which are not suited for annotated interactive non-linear videos are randomized replacement strategies. Because of a high level of knowledge on the structure of the annotated interactive non-linear video, it is apparent that those policies do not accomplish the intended goal.

The two fitting categories of replacement policies are the size-based and the function-based ones. Nevertheless, not all algorithms of these categories make sense for annotated interactive non-linear video. Size-based policies like PSS [AWY99] and its extensions CSS [Tat98] and LRU-SP [CK00] are based on size and use access frequency but in annotated interactive non-linear videos, the access frequency for most of the elements is one. Thus, the results would mainly be based solely on the size of the elements. Their behavior is similar to SIZE [Abr$^+$96] then, which could be used for annotated interactive non-linear video, but provides absolutely no timing information. The more time has elapsed since the element was displayed, the lesser is the relative frequency that it is viewed again. Therefore, the timing information is important. RTIME [FO01] uses the download time which is needed to load the element from the web server into the cache. The used value is that from the last download of the element. This results in a behavior as described for SIZE when it is used in annotated interactive nonlinear videos. Two policies which could be suitable for annotated interactive non-linear video are LRU-Min [Abr$^+$95] and partitioned caching [MAJ98], because they are based on size and time since the last reference.

Function-based policies can be quite different in their functionality. None of them can be applied to annotated interactive non-linear video without modifications. Many of them, like GDSF [Che98], GD [JB01], TSP [YZZ01], LGR [BC08], MIX [NLN98], M-Metric [Wes95], Hybrid [WA97], ARC-H [KK12], and LNC-R-W3 [SSV97] use the access frequency of elements as a main factor in their calculations, which makes the results less convincing for our problem. Server assisted cache replacement [CKR98] and LR-Model [FHH00] produce great calculation overhead because of their complexity. They contain far more logic than is needed for annotated interactive non-linear video because of their well-known structure. GD-Size [CI97] and Bolot/Hoschka's strategy [BH96] would be applicable with smaller changes. The

[1] A broad overview on existing cache replacement policies can be found in [Won06] and [PB03], only a part of them is analyzed in this section. Not mentioned policies are not suitable for annotated interactive non-linear videos.

latter one requires a fitting configuration of the tuning parameters. LRV [RV00] and LUV [Bah+02] are probability-based and could also be used with slight adaptations (in replacing the access frequency by another parameter). SEMALRU [GAGM09] extends LRU with semantics. Thereby, the relation of cached documents to incoming documents is evaluated. Related documents stay stored in the cache, while others tend to get deleted. This behavior is useful for annotated interactive non-linear videos. Furthermore, SEMALRU takes the time of the last access into account which is not relevant for annotated interactive non-linear videos. This policy may be usable for annotated interactive non-linear videos with certain adaptations. Gonzalez-Canete, Casilari, and Trivino-Cabrera carried out a study on how LRU, LFU, LFU-DA [AW97], GD-SIZE, GDSF, and GD perform for different content-types like audio files, images, text, or video. They found out that "there is no a replacement policy that outperforms the others for all content-types, so to develop a proxy cache that distinguishes the content-types of documents, the best algorithm for each content-type should be applied" [GCCTC07].

Caching and retransmission strategies as well as transcoding for multimedia objects on web proxies are described by Li and Ong [LO09], Liu and Li [LL04], Wang et al. [Wan+02], Park et al. [Par+07], Wu, Chong, and Givan [WCG06], Xiang, Zhang, and Zhu [XZZ03], and Liao and Shih [LS02]. These solutions are not applicable for our work, because we are not using a proxy so far. Some aspects of the algorithms used for the proxies might be integrated in our work to provide a more flexible and fine-grained scheduling at client side. They can also be integrated in our overall setup to reduce network load.

Several approaches of caching proxies for television services can be found. Avramova et al. [Avr+11a] propose an algorithm for the caching of catch-up television services. Wauters et al. [Wau+06] and Li and Simon [LS11] describe co-operative proxy caching algorithms for time-shifted IPTV services. Caching algorithms which are tracking the popularity of objects in video on demand and catch-up TV services are evaluated by De Vleeschauwer and Laevens [DVL09]. The pre-fetching and caching of online TV services provided by a hosting service called "hulu" is examined by Krishnappa et al. [Kri+11]. The most popular videos of a week are pre-fetched. This approach is compared to other approaches. Liu et al. [Liu+04] describe a caching strategy which takes the popularity of a cached element into account. Doing that, they achieve low user start-up latency as well as high bandwidth savings. The idea of pre-fetching videos/scenes which are watched with a high probability is used in our work, too, but we decide which elements have to be pre-fetched by the user behavior while watching a video.

A buffer replacement algorithm for interactive media is described by Cho et al. [Cho+03]. Interactivity is limited to jumps (for- and backward) in linear videos. The cache hit value is improved by keeping intervals that are removed by VCR as virtual intervals. They might be accessed with a higher frequency than others. This algorithm can be used to refine our delete scheduling within a scene. A simple caching scheme at client side for interactive video-on-demand is introduced by Branch, Egan, and Tonkin [BET99]. Interactivity is limited to jumps in linear videos, too. If the cache is full, an aging mechanism is used to decide which frames are the most recently requested ones and can be deleted. (This section (4.5 Cache Management and Replacement Strategies) was taken and adapted from our previous work [MH12].)

Critical reflection: Caching proxies or a direct download to the client is proposed in related work. Caching proxies (see Table 4.5) are not intended to be used in this work. Commonly used replacement policies are partially applicable to our cache deletion problem. Recency-

based, frequency-based, and randomized factors cause a contrary effect regarding the availability of already viewed elements in the cache because of their assumptions. From the five categories of cache replacement factors/policies (as defined by Wong [Won06]) only size-based and function-based strategies can be used for annotated interactive non-linear videos with some adaptations. Nevertheless, none of both categories takes the well known structure of annotated interactive non-linear videos into account. All described policies use current values of objects or historic values, but with annotated interactive non-linear videos, predictions on which possible future elements might be needed can be made. This knowledge helps to avoid unnecessary retransmissions of objects known from the server. Using the definition of Aggarwal, Wolf, and Yu, a distinction between direct extensions of traditional policies, key-based policies, and function-based policies is made [AWY99]. Table 4.6 shows the categorization from Wong [Won06] as factors and the categorization of Aggarwal, Wolf, and Yu [AWY99] as policies. It can be noted, that only LRU-Min [Abr+95], partitioned caching [MAJ98], GD-Size [CI97], Bolot/Hoschka's strategy [BH96], LRV [RV00], LUV [Bah+02], and SEMALRU [GAGM09] can be used for annotated interactive non-linear video. Adaptations are necessary for all of them, because none of them uses structural information.

	Caching	Scheduling	QoS	Streaming	Transcoding	Transmission costs	Replacement	Interactivity	**Suitability for IAV**
Avramova et al. [Avr+11a]	✓	–	–	✓	–	–	✓	–	–
Branch et al. [BET99]	✓	–	–	✓	–	–	–	✓	–
Cho et al. [Cho+03]	✓	–	–	✓	–	–	✓	✓	✓
De Vleeschauwer and Laevens [DVL09]	✓	–	–	✓	–	–	✓	–	–
Krishnappa et al. [Kri+11]	✓	–	–	✓	–	–	✓	–	–
Li and Ong [LO09]	✓	–	✓	✓	✓	✓	–	–	–
Li and Simon [LS11]	✓	–	–	✓	–	–	✓	–	–
Liao and Shih [LS02]	✓	–	–	✓	–	–	✓	✓	–
Liu and Li [LL04]	✓	✓	✓	✓	–	–	–	–	–
Liu et al. [Liu+04]	✓	–	–	✓	–	–	✓	✓	–
Park et al. [Par+07]	✓	–	–	✓	✓	–	✓	–	–
Wang et al. [Wan+02]	✓	✓	–	✓	–	✓	–	–	–
Wauters et al. [Wau+06]	✓	–	–	✓	–	–	✓	✓	–

Table 4.5.: Implementations of streaming proxies with cache replacement algorithms.

4.6. Summary

This section gives an overview of related work from areas concerning our download and cache management. We first outlined three different **implementations of non-linearity**: "real" non-linearity, non-linearity by linking, and non-linearity by time leaps. Only "real" non-linearity is suitable for annotated interactive non-linear videos as defined in this thesis. It gives the impression of one single video and allows the realization of pre-loading algorithms because

	Factor						Policy			Others			
	Recency	Frequency	Size	Cost	Randomization	Prediction	Direct	Key-based	Function-based	Partitioning of cache	Transmission delay	Server communication	**Suitability for IAV**
ARC-H [KK12]	✓	–	–	✓	–	–	–	✓	–	–	–	–	–
CSS [Tat98]	✓	✓	✓	–	–	–	–	✓	–	–	–	–	–
LRU-Min [Abr$^+$95]	✓	–	✓	–	–	–	–	✓	–	–	–	–	✓
LRU-SP [CK00]	✓	✓	✓	–	–	–	–	✓	–	–	–	–	–
partitioned caching [MAJ98]	✓	–	✓	–	–	–	–	✓	–	✓	–	–	✓
PSS [AWY99]	✓	–	✓	–	–	–	–	✓	–	–	–	–	–
RTIME [FO01]	–	–	✓	✓	–	–	–	✓	–	–	–	–	–
SIZE [Abr$^+$96]	✓	–	✓	–	–	–	✓	–	–	–	–	–	–
Bolot/Hoschka [BH96]	✓	–	✓	–	–	✓	–	–	✓	–	✓	–	✓
GD [JB01]	–	✓	✓	✓	–	–	–	–	✓	–	–	–	–
GD-Size [CI97]	–	–	✓	✓	–	–	–	–	✓	–	–	–	✓
GDSF [Che98]	–	✓	✓	✓	–	✓	–	–	✓	–	–	–	–
Hybrid [WA97]	–	✓	✓	–	–	–	–	–	✓	–	✓	–	–
LGR [BC08]	✓	✓	✓	–	–	–	–	–	✓	–	–	–	–
LNC-R-W3 [SSV97]	–	–	✓	–	–	–	–	–	✓	–	✓	–	–
LR-Model [FHH00]	✓	–	✓	✓	–	✓	–	–	✓	–	–	–	–
LRV [RV00]	✓	✓	✓	–	–	–	–	–	✓	–	–	–	✓
LUV [Bah$^+$02]	✓	✓	–	✓	–	–	–	–	✓	–	–	–	✓
M-Metric [Wes95]	✓	✓	✓	–	–	–	–	–	✓	–	–	✓	–
MIX [NLN98]	✓	✓	✓	–	–	–	–	–	✓	–	✓	–	–
SEMALRU [GAGM09]	✓	–	–	–	–	–	–	–	✓	–	–	–	✓
Server assisted c. repl. [CKR98]	–	–	✓	✓	–	✓	–	–	✓	–	–	✓	–
TSP [YZZ01]	–	✓	✓	✓	–	✓	–	–	✓	–	–	–	–

Table 4.6.: Basic and extended replacement policies for caches.

of the usage of separate video scenes. Furthermore, scenes can be used more than once and extended with different annotations.

In a second step, we analyzed how related work deals with **user behavior** when VCR functions, non-linearity, and interactivity are provided in videos. We found two groups of related work. One had a focus on VCR action, the other on interactivity or non-linearity. The two most relevant works were those from Fei et al. [Fei$^+$05], who propose a sliding window for buffer management and a user behavior model, and Grigoras, Charvillat, and Douze [GCD02], who use an optimized Markov Decision Process to optimize the pre-fetching of hypervideos to reduce network transmission latency.

Scheduling algorithms are needed to insert elements from different scenes into one or more download queues. A suitable algorithm should be efficient in creating queues, able to weight queues, and not rearrange already created queues (but take the first element from a queue). An analysis of 18 scheduling algorithms reveals eight suitable algorithms from which we chose DRR [SV95] because of its very good applicability to our scheduling problem with easy adaptations.

We also analyzed work from the area of **download and streaming** of interactive (non-linear) videos. These works deal with different problems which arise when videos are transmitted over networks. Some focus on the streaming of videos and thereby existing network or cache issues, others deal with pre-fetch and scheduling of data. A major goal is to provide a certain level in quality of service or quality of experience. None of the proposed algorithms and methods can be applied directly on our research problem, but some of them may be used to improve and refine our approach in future work.

Finally, we evaluated **cache management and replacement strategies**. Caching proxies cannot be used in our work. Thus we focused more on replacement policies for web caches. Recency-based, frequency-based, and randomized factors cannot be used in our work because they cause a contrary effect regarding the availability of already viewed elements in the cache. From the five categories of cache replacement factors/policies (as defined by Wong [Won06]) only size-based and function-based strategies can be used for annotated interactive non-linear videos with some adaptations. None of the found traditional cache replacement strategies takes the well known structure of annotated interactive non-linear videos into account, which can be used to make predictions on elements that might be needed in the future playback. Some of the found algorithms can be used, but adaptations are necessary for all of them for that reason.

5. Formalized Video Model, Hardware Constraints, and User Behavior

A formalized video model describes the structure of an annotated interactive non-linear video. It describes sequences of scenes and the point in time at which the viewer can interact with the video, as well as the points in time where annotations are displayed or hidden. Furthermore, it formalizes the described relationships and outlines how the structure of an annotated interactive non-linear video can be formalized in a video model. The playback of an annotated interactive non-linear video is influenced by hardware limitations and the user behavior. Hardware limitations are the cache size at client side and the transmission bandwidth from the client to the server. In this section, we give universal definitions and descriptions for the video model, hardware constraints, and the user behavior. Furthermore, we provide textual descriptions with the definitions. Illustrations are provided for more complex contexts.

Figure 5.1 shows an exemplary structure of an annotated interactive non-linear video representing a tour through the ground floor of a house. The lower left corner of the image depicts the layout of the ground floor. The video scenes are filmed paths through rooms of the house, from one door to another. Viewers are asked where they want to go at certain points and are able to choose their own unique way through the house. The structure of the scenes defines a scene graph. The 16 scenes of the video are represented as labeled rectangles. The rhombus symbolizes a fork in the flow where the viewer can choose a scene. Possible targets of a scene are other scenes, a fork, or the end of the video. The decision, which path is followed, depends on the click of the appropriate button. The scene graph has a source (start) and a sink (end). It is directed, weighted, and possibly cyclic. Individual scenes are annotated with detailed images of furniture or flooring. Text-annotations, images, audio files, and videos describe room specifics or items shown in a scene that reach beyond the information provided by the video. Images provide detailed views of objects in the video. Figure 5.2 shows the time spans for displaying annotations of the entrance scene in a detailed view. This scene consists of a video with 710 frames (green boxes). During the playback of the video six annotations are shown and hidden as exemplified by the blue boxes. (Parts of this section were taken and adapted from our previous work [MH12].)

5.1. Video Model

A video model describes the internal structure of the annotated interactive non-linear video. A projection function and other auxiliary functions are needed in the remainder of this work as well as a generalization from frames and annotations to downloadable elements and other sets. The definitions from this section are necessary to develop algorithms and strategies for downloading and caching.

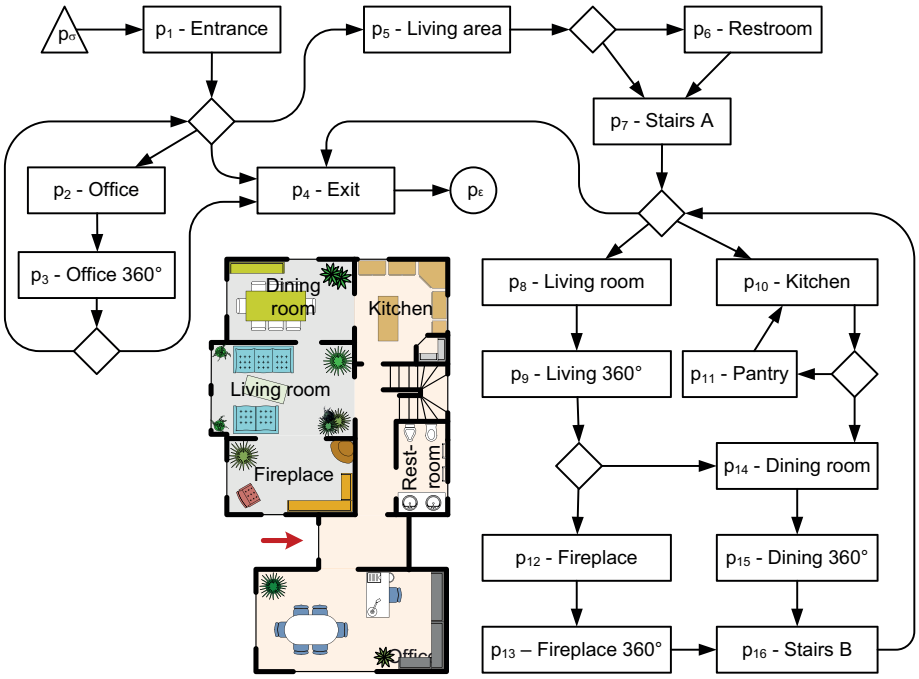


Figure 5.1.: Exemplary scene graph of a tour through the ground floor of a house with six rooms.

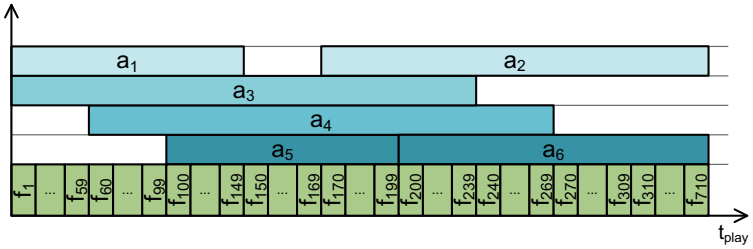


Figure 5.2.: Exemplary schedule for six annotations of one scene in detail

5.1.1. Annotated Interactive Non-linear Video

We now give a formal definition of all elements used during playback as well as the relationships between the individual elements. Furthermore, control information is defined which is needed during playback to select the successor scene at a fork. The individual definitions are used to formulate the definition of “annotated interactive non-linear video”.

Frames and annotations are fundamental elements of an annotated interactive non-linear video. A frame consists of a sequence of bits as defined in Definition 5.1. An annotation contains a media object α (see Definition 5.2) and has a priority (see Definition 5.3). The media object may be any type of additional content like text, image, video, or audio files and is, like the frame, defined as a sequence of bits. These are stored on the client hardware and have to be transmitted over the network. The content of an annotation is always downloaded as a single block whether it is a continuous or a static medium. More precisely, an annotation a is a pair of the media object α and a priority Λ (see Definition 5.4).

Definition 5.1 (*Frame f*)

A frame f is an n-tuple of bits representing an image; $f := (\chi_1, \ldots, \chi_n) \in \{0,1\}^n, n \in \mathbb{N}^+$.

Definition 5.2 (*Content of an Annotation α*)

The content of an annotation α is an n-tuple of bits representing a media object; $\alpha := (\chi_1, \ldots, \chi_n) \in \{0,1\}^n, n \in \mathbb{N}^+$.

Definition 5.3 (*Priority of an Annotation Λ*)

The priority of an annotation is Λ, $\Lambda \in \mathbb{N}^+$. The higher Λ is, the lower is the priority of the annotation.

Definition 5.4 (*Annotation a*)

An annotation $a := (\alpha, \Lambda),\ \Lambda \in \mathbb{N}^+$ is a pair of the content of the annotation α and the priority of the annotation Λ.

The previously described elements can be combined to sets and tuples for further calculations and definitions. We therefore define the set of frames $\mathcal{F}_\mathcal{V}$ (see Definition 5.5) and the set of annotations $\mathcal{A}_\mathcal{V}$ (see Definition 5.6) of an annotated interactive non-linear video.

Definition 5.5 (*Set of Frames $\mathcal{F}_\mathcal{V}$*)

$\mathcal{F}_\mathcal{V}$ is a finite set of frames of the annotated interactive non-linear video $\mathcal{V}$.

Definition 5.6 (*Set of Annotations $\mathcal{A}_\mathcal{V}$*)

$\mathcal{A}_\mathcal{V}$ is a finite set of annotations of the annotated interactive non-linear video $\mathcal{V}$.

An n-tuple containing pairs of a frame and a set of annotations is representing a scene p (see Definition 5.7). The set of annotations attached to the frame indicates that all annotations in the set are displayed with the frame. An annotated interactive non-linear video furthermore has a start scene p_σ and an end scene p_ϵ with just one frame and an empty set of annotations (see Definitions 5.8 and 5.9). All scenes can be combined to a set of scenes $\mathcal{P}_\mathcal{V}$ (see Definition 5.10).

Definition 5.7 (*Scene p*)

A scene p is an n-tuple of pairs each containing a frame and a set of annotations which are displayed with the frame; $p_x := ((f_{x,1}, A_{x,1}), \ldots, (f_{x,n}, A_{x,n})),\ x, n \in \mathbb{N}^+, f_{x,i} \in \mathcal{F}_\mathcal{V}, A_{x,i} \subseteq \mathcal{A}_\mathcal{V}, 1 \leq i \leq n$

Definition 5.8 (*Start Scene p_σ*)

The start scene p_σ is a 1-tuple containing a pair representing a single frame without an annotation; $p_\sigma := ((f_{\sigma,1}, \{\}))$.

Definition 5.9 (*End Scene p_ϵ*)

The end scene p_ϵ is a 1-tuple containing a pair representing a single frame without an annotation; $p_\epsilon := ((f_{\epsilon,1}, \{\}))$.

Definition 5.10 (*Set of Scenes $\mathcal{P}_\mathcal{V}$*)

The set of scenes $\mathcal{P}_\mathcal{V}$ of the annotated interactive non-linear video $\mathcal{V}$ is defined as $\mathcal{P}_\mathcal{V} := \{p_\sigma, p_1, \ldots, p_x, p_\epsilon\},\ x \in \mathbb{N}^+$

The whole annotated interactive non-linear videos can be described with the elements and sets defined so far as a deterministic finite state machine $\mathcal{N}_\mathcal{V}$ (see [Ill09, p. 14 et seq.]). Thereby, the input symbols are defined as a set of Boolean functions as defined in [Got01, p. 40]. Restrictions need to be applied for the start and the end scene.

Definition 5.11 (*Annotated Interactive Non-linear Video* $\mathcal{V}$)

An annotated interactive non-linear video $\mathcal{V}$ *is defined as a deterministic finite state machine* $\mathcal{N}_\mathcal{V} := (\mathcal{P}_\mathcal{V}, \Sigma, \delta, p_\sigma, \{p_\epsilon\})$ *with*
$\Sigma := \{w_{i,j} | w_{i,j}$ *is a button triggering the selection of a successor scene,* $i \in \{1, \ldots, |\mathcal{P}_\mathcal{V}| - 2, \sigma\}, j \in \{1, \ldots, |\mathcal{P}_\mathcal{V}| - 2, \epsilon\}\}$ *and* $\delta : \mathcal{P}_\mathcal{V} \times \Sigma \to \mathcal{P}_\mathcal{V}$.
The following restrictions are applied: $\exists! k : \delta(p_\sigma, w_{\sigma,k}) \to p_k \ \wedge \ \nexists k : \delta(p_k, w_{k,\sigma}) \to p_\sigma \ \wedge \ \exists k : \delta(p_k, w_{k,\epsilon}) \to p_\epsilon \ \wedge \ \nexists k : \delta(p_\epsilon, w_{\epsilon,k}) \to p_k$

The deterministic finite state machine $\mathcal{N}_\mathcal{V}$ defines possible successors of a scene and which buttons have to be clicked to access a designated successor scene. The transition $(p_m, w_{i,j}) \to p_n \in \delta$ implies that scene p_n is successor of scene p_m. When the button clicks are logged, relative frequencies can be applied to transitions based on the previous user behavior. The cold start problem can be overcome by applying a default value based on a probability distribution function to all successor scenes of a scene. For a more detailed description see Section 6.4.3.

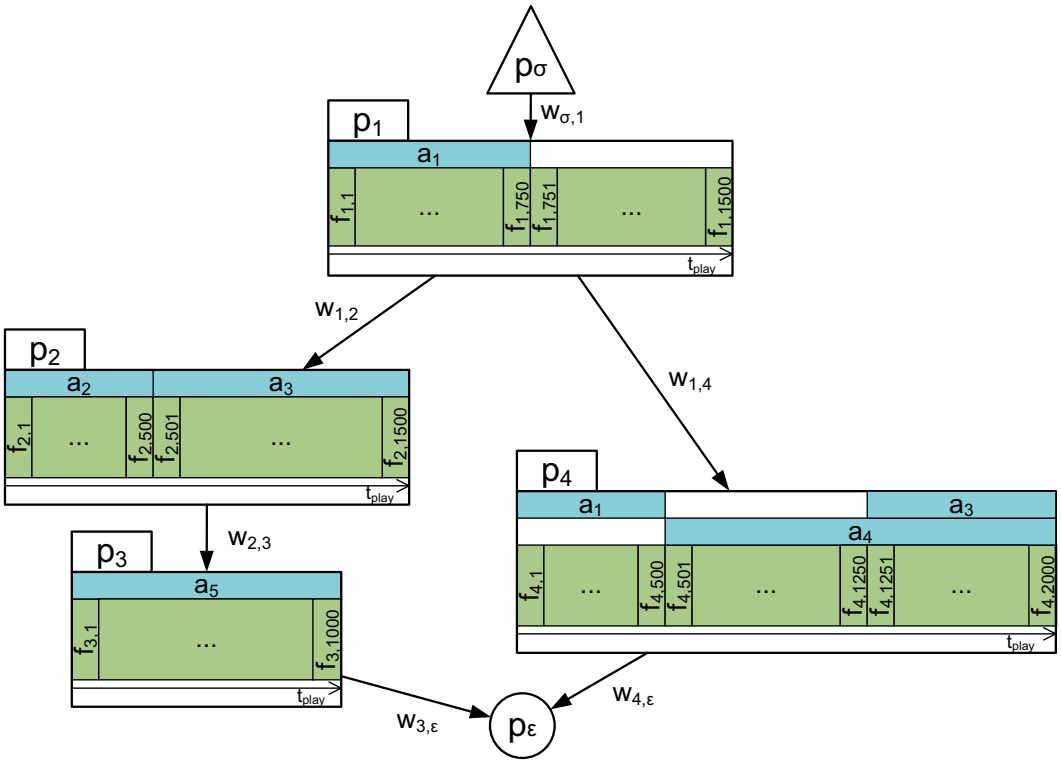


Figure 5.3.: Example of an annotated interactive non-linear video with six scenes (including start and end scene) and five annotations.

Definition 5.11 will now be illustrated with a small example as shown in Figure 5.3. This annotated interactive non-linear video has six scenes, including start and end scene, and five annotations. The **set of scenes** is defined as $\mathcal{P}_\mathcal{V} = \{p_\sigma, p_1, p_2, p_3, p_4, p_\epsilon\}$. The different scenes can be described as follows:

- $p_\sigma = ((f_{\sigma,1}, \{\}))$
- $p_1 = ((f_{1,1}, \{a_1\}), \ldots, (f_{1,750}, \{a_1\}), (f_{1,751}, \{\}), \ldots, (f_{1,1500}, \{\}))$
- $p_2 = ((f_{2,1}, \{a_2\}), \ldots, (f_{2,500}, \{a_2\}), (f_{2,501}, \{a_3\}), \ldots, (f_{2,1500}, \{a_3\}))$
- $p_3 = ((f_{3,1}, \{a_5\}), \ldots, (f_{3,1000}, \{a_5\}))$
- $p_4 = ((f_{4,1}, \{a_1\}), \ldots, (f_{4,500}, \{a_1\}), (f_{4,501}, \{a_4\}), \ldots, (f_{4,1250}, \{a_4\}),$ $(f_{4,1251}, \{a_3, a_4\}), \ldots, (f_{4,2000}, \{a_3, a_4\}))$
- $p_\epsilon = ((f_{\epsilon,1}, \{\}))$

The **set of frames** is set to $\mathcal{F}_\mathcal{V} = \{f_{\sigma,1}, f_{1,1}, \ldots, f_{1,1500}, f_{2,1}, \ldots, f_{2,1500}, f_{3,1}, \ldots, f_{3,1000}, f_{4,1}, \ldots, f_{4,2000}, f_{\epsilon,1}\}$ and contains 6002 frames which are divided up into the six scenes. The first and second scene, p_1 and p_2, each consist of 1500 frames, the third scene p_3 consists of 1000 frames and the fourth scene p_4 consists of 2000 frames. The **set of annotations** contains five elements. It is defined as $\mathcal{A}_\mathcal{V} = \{a_1, a_2, a_3, a_4, a_5\}$.

The **transition function** δ defines where and under what conditions transitions from one scene to another are allowed. The transition $\delta(p_\sigma, w_{\sigma,1}) \rightarrow p_1$ sets the first scene of the video. Transitions $(p_3, w_{3,\epsilon}) \rightarrow p_\epsilon$ and $(p_4, w_{4,\epsilon}) \rightarrow p_\epsilon$ indicate two different last scenes of the video followed by the end scene. A linear transition is also defined from scene p_2 to p_3 with $\delta(p_2, w_{2,3}) \rightarrow p_3$. In these cases the follow-up scenes start immediately after the predecessor scenes end. The remaining two transitions, $\delta(p_1, w_{1,2}) \rightarrow p_2$ and $\delta(p_1, w_{1,4}) \rightarrow p_4$, describe a selection panel at the end of scene p_1. The viewer in this example selects button $w_{1,2}$ or button $w_{1,4}$. Only one of the buttons/paths can be selected. (This subsection (5.1.1 Annotated Interactive Non-linear Video) was taken and adapted from our previous work [MH12].)

5.1.2. Basic Functions and Definitions

Further definitions and various basic functions are useful for the calculations in the remainder of this work, which will be defined hereafter. Thereby, $\mathcal{X} \in \{\mathcal{A}_\mathcal{V}, \mathcal{F}_\mathcal{V}\}$.

The **frame rate** of the video may either be a constant or variable (as described in [CH97; Kim+00; Pan+04; Shu+93]). A constant frame rate is defined in Function 5.1. It is set to a fixed value c_r for all calculations in this work. Usually c_r is set to 25 or 30 fps.

$$r : \mathbb{R}^+ \mapsto \mathbb{N}^+, t \mapsto r(t) := c_r \tag{5.1}$$

A **dimension function** is needed to get the size/length of a tuple. This basic function is defined in Function 5.2.

$$dim : \mathcal{X}^k \mapsto \mathbb{N}^+, (x_1, \ldots, x_k) \mapsto dim(x_1, \ldots, x_k) := k \tag{5.2}$$

A **projection function** is needed to get a specific value from a tuple. This basic function is defined in Function 5.3.

$$\pi_i : \mathcal{X}^k \mapsto \mathcal{X}, k \in \mathbb{N}^+, (x_1, \ldots, x_k) \mapsto \pi_i(x_1, \ldots, x_k) := x_i,\ 1 \leq i \leq k \tag{5.3}$$

A second **projection function** can be used to get a part of a tuple. This basic function is defined in Function 5.4.

$$\pi_{i_1,i_2} : \mathcal{X}^j \mapsto \mathcal{X}^k, j,k \in \mathbb{N}^+, k \leq j, (x_1,\ldots,x_j) \mapsto \pi_{i_1,i_2}(x_1,\ldots,x_j) := (x_{i_1},\ldots,x_{i_2}), \quad 1 \leq i_1 < i_2 \leq j, i_2 - i_1 + 1 = k \tag{5.4}$$

Furthermore, a generalization from frames and annotations to "downloadable elements" simplifies some of our calculations. The annotations $\mathcal{A}_\mathcal{V}$ and the frames $\mathcal{F}_\mathcal{V}$ of a video $\mathcal{V}$ are joined to a **set of (downloadable) elements** of a video $\mathcal{E}_\mathcal{V}$ (see Definition 5.12).

Definition 5.12 (*Set of Downloadable Elements $\mathcal{E}_\mathcal{V}$ of a Video*)

$\mathcal{E}_\mathcal{V}$ is a set of downloadable elements of a video $\mathcal{V}$, which is defined as the union $\mathcal{E}_\mathcal{V} = \mathcal{A}_\mathcal{V} \cup \mathcal{F}_\mathcal{V}$ of the set of frames $\mathcal{F}_\mathcal{V}$ and the set of annotations $\mathcal{A}_\mathcal{V}$ of the video.

A **size function** is defined in Function 5.5. It returns the size of an element by returning the length of the n-tuple of bits representing the content of the frame or annotation. This function is needed to get the amount of data that has to be downloaded from the server or has to be stored in the cache for each frame or annotation.

$$s : \mathcal{E}_\mathcal{V} \to \mathbb{N}^+, e_i \mapsto s(e_i) := \begin{cases} dim(e_i) & \text{if } e_i \text{ is a frame} \\ dim(\pi_1(e_i)) & \text{if } e_i \text{ is an annotation} \end{cases} \tag{5.5}$$

Function 5.6 returns the **priority** of an annotation by a projection on the second component of the annotation pair (α_o, Λ). The higher the priority is, the lower is its number. The highest priority is "1". If no priorities are used, all annotations are set to priority "1" and are treated with the same priority with which frames are downloaded.

$$q : \mathcal{A}_\mathcal{V} \to \mathbb{N}^+, a_o \mapsto q(a_o) := \pi_2(a_o) = \Lambda \tag{5.6}$$

The **duration of a scene** p_x in seconds $l(p_x)$ is calculated by the division of the number of frames of the scene $dim(p_x)$ by the frame rate r at a fixed frame rate. This is expressed by Function 5.7.

$$l : \mathcal{P}_\mathcal{V} \to \mathbb{N}^+, p_x \mapsto l(p_x) := \frac{dim(p_x)}{c_r} \tag{5.7}$$

A **tuple of frames, the set of frames, and the set of annotations of a scene** simplify the design of formulas for calculations within a scene. The tuple $\widehat{p}_x^f$ contains all frames of scene p_x (see Definition 5.13). The set $\mathcal{F}_{p_x}$ contains all frames of scene p_x (see Definition 5.14). The tuple $(f_{x,1},\ldots,f_{x,n})$ may be simplified to $(f_1,\ldots,f_n)$, if it is apparent from the context that only one scene is regarded. Set $\mathcal{A}_{p_x}$ includes all annotations linked to frames of scene p_x (see Definition 5.15).

Definition 5.13 (*Tuple of Frames of a Scene $\widehat{p}_x^f$*)

The tuple of frames $\widehat{p}_x^f$ of a scene p_x is the projection on the first component of each pair, $\widehat{p}_x^f := (\pi_1((f_{x,1},A_{x,1})),\ldots,\pi_1((f_{x,n},A_{x,n}))) = (f_{x,1},\ldots,f_{x,n}),\ x,n \in \mathbb{N}^+, f_{x,i} \in \mathcal{F}_\mathcal{V}, 1 \leq i \leq n$

Definition 5.14 (*Set of Frames of a Scene $\mathcal{F}_{p_x}$*)

The set of frames $\mathcal{F}_{p_x}$ of a scene p_x is the projection on each component of the tuple of frames of a scene $\widehat{p}_x^f$, $\mathcal{F}_{p_x} := \{\pi_1(\widehat{p}_x^f), \ldots, \pi_n(\widehat{p}_x^f)\} = \{f_{x,1}, \ldots, f_{x,n}\},\ x,n \in \mathbb{N}^+, f_{x,i} \in \mathcal{F}_\mathcal{V}, 1 \leq i \leq n$

Definition 5.15 (*Set of Annotations of a Scene $\mathcal{A}_{p_x}$*)

The set of annotations $\mathcal{A}_{p_x}$ of a scene p_x is union of the set of the projections on the second component of each pair, $\mathcal{A}_{p_x} := \pi_2((f_{x,1}, A_{x,1})) \cup \ldots \cup \pi_2((f_{x,n}, A_{x,n})) = A_{x,1} \cup \ldots \cup A_{x,n} = \{a_1, \ldots, a_j\},\ x,j \in \mathbb{N}^+, a_i \in \mathcal{A}_\mathcal{V}, 1 \leq i \leq j$

Furthermore, a generalization from frames and annotations of a scene to "downloadable elements of a scene" can be defined as the join of the annotations $\mathcal{A}_{p_x}$ and the frames $\mathcal{F}_{p_x}$ of a scene p_x to downloadable elements of a scene $\mathcal{E}_{p_x}$ (see Definition 5.16). We use j_i instead of $f_{x,i}$ if we describe algorithms or functions for one scene in the remainder of this work.

Definition 5.16 (*Set of Downloadable Elements $\mathcal{E}_{p_x}$ of a Scene*)

$\mathcal{E}_{p_x}$ is a set of downloadable elements of a scene p_x, which is defined as the union $\mathcal{E}_{p_x} = \mathcal{A}_{p_x} \cup \mathcal{F}_{p_x}$ of the set of frames $\mathcal{F}_{p_x}$ and the set of annotations $\mathcal{A}_{p_x}$ of the scene.

The **set of all successor scenes of a scene** $\mathcal{P}_{succ(p_x)}$ is needed for several calculations concerning the download as well as the selection of a scene at a fork (see Definition 5.17).

Definition 5.17 (*Set of Successor Scenes $\mathcal{P}_{succ(p_x)}$*)

The set successor scenes $\mathcal{P}_{succ(p_x)}$ of scene p_x is defined as $\mathcal{P}_{succ(p_x)} := \left\{p_n | \exists w_{x,n} : \delta(p_x, w_{x,n}) \rightarrow p_n\right\}$

(Parts of this subsection (5.1.2 Basic Functions and Definitions) were taken and adapted from our previous work [MH12].)

5.2. Hardware Constraints

Hardware constraints are induced by hardware limitations at client side, at server side, or during data transmission from client to server. Limitations at client side may be a restricted browser cache size, a slow network connection, or a low speed graphics adapter used for decoding. Slow network connections or many concurrent connections have a bearing on the transmission speed at server side. Furthermore, the network connection between client and server may be limited.

5.2.1. Cache Size

The maximum available cache size of the client is called B and is defined as the time-dependent Function 5.8. It is given by the end-user device and we assume it is a static value c_B for the whole duration of the video.

$$B : \mathbb{N}^+ \rightarrow \mathbb{N}^+, t \mapsto B(t) := c_B \tag{5.8}$$

We conduct our experiments with a fixed cache size and are able to make statements about the average case assuming that c_B is the value for the average cache size, or the worst case assuming that c_B is the minimum cache size. We furthermore use one single cache in which elements are stored, no predefined division of the cache size is intended due to the variety in the composition of elements of which annotated interactive non-linear videos may exist. In future work, different time-pendent functions for the cache size may be tested.

5.2.2. Bandwidth

The transmission bandwidth BW is defined as a time-dependent function in Function 5.9. We assume that the bandwidth is a constant value c_{BW} during the whole transmission time. The effective bandwidth for data transmission is the smaller of server, client, and medium bandwidth, as expressed by Equation 5.10.

$$BW : \mathbb{N}^+ \rightarrow \mathbb{N}^+, t \mapsto BW(t) := c_{BW} \tag{5.9}$$

$$c_{BW} := \min\{c_{BW}^{client}, c_{BW}^{server}, c_{BW}^{medium}\} \tag{5.10}$$

Statements about the average and the worst case can be made, if we assume that c_{BW} is the average or the minimum bandwidth. We assume that our values represent the effective bandwidth. Delays which result from retransmissions, packet loss, or other network related characteristics are not taken into account. (Parts of this section (5.2 Hardware Constraints) were taken and adapted from our previous work [MH12].)

5.3. User Behavior

Different means of navigation are offered to the user. In-scene and in-video navigation need to be distinguished. The viewer is for example able to perform standard VCR interaction during a scene. Performing in-video navigation, for example fast-backward may be possible by going back in the history of already watched scenes. Fast-forward may be allowed, too, but it only works with a linear sequence of scenes. The user has to make a selection at a fork in the video flow. Allowed user interactions need to be specified. Probabilities can be assigned to single actions or combinations of actions as described by Fei et al. in their user behavior model [Fei$^+$99]. This model provides the probability of a specific user interaction as well as its mean duration. These values can then be used for calculations in the download and cache management algorithms and strategies. Probabilities or relative frequencies may be used as a factor to prioritize elements for download (see Section 6.4). Durations for user interaction enhance the available times for downloads during playback. Designated characteristics of a user behavior model may be combined to usage patterns which can then be evaluated for a group of annotated interactive non-linear videos. Hereafter we describe possible VCR actions and interactive elements.

5.3.1. VCR Actions

Besides play, pause, and stop, several other VCR actions are possible. It is also possible to play the video backwards with the frame rate used for playing it forward. Besides slow- and fast-forward or rewind, it is furthermore possible to jump to a certain frame forward or backward in the currently played scene. Table 5.1 enlists all actions which may be considered in this work in a single scene p_x. The indices of the scenes are simplified from $f_{x,i}$ to f_i for that reason.

Action	Current frame	Current frame rate	New frame	New frame rate	Remark
Stop	f_m	c_r	f_σ	0	
Pause	f_m	c_r	f_m	0	
Play forward	f_m	0	f_{m+1}	c_r	f_m not last frame of a scene
	f_m	c_r	f_{m+1}	c_r	f_m not last frame of a scene
Play backward	f_m	0	f_{m-1}	c_r	f_m not first frame of a scene
	f_m	c_r	f_{m-1}	c_r	f_m not first frame of a scene
Jump forward	f_m	c_r	f_{m+x}	c_r	$x \in \mathbb{N}^+, f_m, f_{m+x}$ in same scene
Jump backward	f_m	c_r	f_{m-x}	c_r	$x \in \mathbb{N}^+, f_m, f_{m-x}$ in same scene
Slow forward	f_m	c_r	f_{m+1}	$c_{r_{SFW}}$	$c_r < c_{r_{SFW}}$, f_m not last frame of a scene
Slow rewind	f_m	c_r	f_{m-1}	$c_{r_{SBW}}$	$c_r < c_{r_{SBW}}$, f_m not first frame of a scene
Fast-forward	f_m	c_r	f_{m+1}	$c_{r_{FFW}}$	$c_r < c_{r_{SFW}} < c_{r_{FFW}}$, f_m not last frame of a scene
Fast rewind	f_m	c_r	f_{m-1}	$c_{r_{FBW}}$	$c_r < c_{r_{SBW}} < c_{r_{FFW}}$, f_m not first frame of a scene

Table 5.1.: Different intra-scene VCR actions.

5.3.2. Extended Interactivity and Navigation

Annotated interactive non-linear videos provide additional features besides the already described VCR actions. The following facts are summarized in Table 5.2. We assume, that the selection panels or quizzes are usually displayed after a scene ends. The user decides which scene should be displayed next either by selecting it directly in a button panel or by solving a quiz. In the latter case, the follow-up scene is chosen by the score reached in the quiz. Each score is assigned to a point range of a scene which is then selected accordingly. Jumps in the whole video which are not structure dependent are selections in a table of contents or a selection in search results. When a user opens the table of contents, the video may stop and continue playing after a user selection, or it may continue playing, depending on the positioning of the table of contents (side area or overlay). The selected entry starts the playback of a scene at its beginning. A search is usually carried out during the playback of a scene. As described in Section 3.1, it is possible to jump to the beginning of a scene or to an annotation in a scene. Interactive functions like pan, tilt, and zoom have no influence on the order of the displayed frames or the frame rate. They may rather increase the download volume, because

higher resolutions of single frames or other camera positions are needed at client side. (Parts of this section were taken and adapted from our previous work [MH12].)

Action	Current frame	Current frame rate	New frame	New frame rate	Remark
Selection/quiz*	$f_{a,m}$	0	$f_{b,1}$	c_r	$f_{a,m}$ last frame of a scene, $f_{b,1}$ first frame of selected successor scene
TOC*	$f_{a,m}$	c_r	$f_{a,m}$	0	user interaction to invoke table of contents at frame $f_{a,m}$
	$f_{a,m}$	0	$f_{a,1} \vee f_{b,1}$	c_r	selection in overlay table of contents at frame $f_{a,m}$, jump to first frame $f_{a,1}$ of same scene or $f_{b,1}$ of other selected scene; thereby, the video pauses
	$f_{a,m}$	c_r	$f_{a,1} \vee f_{b,1}$	c_r	selection in side area table of contents at frame $f_{a,m}$, jump to first frame $f_{a,1}$ of same scene or $f_{b,1}$ of other selected scene
Keyword search*	$f_{a,m}$	c_r	$f_{a,m}$	0	user interaction to invoke keyword search at frame $f_{a,m}$
	$f_{a,m}$	0	$f_{a,k} \vee f_{b,k}$	c_r	select annotation in search results at frame $f_{a,m}$, jump to frame $f_{a,k}$ of same scene or $f_{b,k}$ of other scene where selected annotation is displayed
	$f_{a,m}$	0	$f_{a,1} \vee f_{b,1}$	c_r	select scene in search results at frame $f_{a,m}$, jump to first frame $f_{a,1}$ of same scene or $f_{b,1}$ of other scene where selected annotation is displayed
Pan/tilt/zoom	$f_{a,m}$	c_r	$f_{a,m+1}$	c_r	user interaction to modify the presentation of the following frames

Table 5.2.: Interactive and navigational actions which are possible in a single scene (intra-scene) or in between scenes (inter-scene) (the * denotes actions which are not limited to a single scene).

5.4. Summary

In this section we propose basic definitions and functions. Furthermore, we assume that certain values are constants and describe limitating circumstances. We first describe a **video model** and define the term "annotated interactive non-linear video". Further definitions include basic functions for the frame rate, and the size and priority of elements. Furthermore, two projection functions, a distance function, and a dimension function are defined for operations on the tuples. Useful sets are the set of all frames and annotations of a single scene, the set of (downloadable) elements, and the set of all successor scenes of a scene. The second

part of this section defines **hardware constraints**, which are the cache size and the available transmission bandwidth. Finally, we describe possible **user interactions**. Thereby we distinguish between VCR actions, like play, stop, pause, fast-forward, and rewind, and extended interactivity and navigation, like selections, a table of contents, a keyword search, and pan, tilt, or zoom.

6. Download and Cache Management

Dealing with a linear video, the estimated download or buffer duration can be calculated for a given bandwidth. After that, the point from which the video can be played without interruption in the video flow can be computed. In contrast, the structure of annotated interactive non-linear videos may lead to problems during playback if the videos cannot be downloaded to the playback device as a whole. Breaks in the video flow after user interactions destroy the perception of a single video. Breaks may occur as a result of loading times for new contents. Concerning one single video, Hossfeld et al. show that "“interruptions [during playback] have to be avoided in any case, even at costs of increased initial delays for filling up the video buffers” [Hos+12]. They “observe a clear preference of initial delays instead of stalling” [Hos+12] in their tests. They furthermore observe “that about 10% of the users do in fact prefer stalling. A possible explanation may be uncertainty and discomfort whether the service is working or not, while service interruptions give a clear feedback. Nevertheless, almost all users prefer uninterrupted service” [Hos+12]. “Regarding initial delays, users learn from everyday usage of applications how much waiting time can be expected, independent of the duration of the service consumption period afterwards. In contrast to initial delays, stalling invokes a sudden, unexpected service interruption. Hence, recency effects apply and impact QoE” [Hos+12]. Hossfeld et al. furthermore found out, that “users tolerate one stalling event per clip as long as stalling event duration remains below 3 s” [Hos+11]. These findings are specified by Egger at al., who describe that “users tend to be highly dissatisfied with two or more stalling events per clip. However, for the case of a stalling length of one second, the user ratings are substantially better for same number of stallings. Nonetheless, users are likely to be dissatisfied in case of four or more stalling events, independent of stalling duration” [Egg+12]. They in addition found out that “users learn from everyday interaction with an application how much waiting time is expected e.g. when logging in to a social network. Furthermore, the duration of the task itself may also influence the experience” [Egg+12]. Besides, “for stalling the video duration matters. In contrast to initial delays, stalling invokes a service interruption by definition. This leads to clearly noticeable disturbance, i.e. a ’bad quality’ event, to which the recency effect applies” [Egg+12]. They furthermore describe that “service interruptions have to be avoided in any case from a user-centric point of view. Even very short stalling events of a few seconds already decrease user perceived quality significantly” [Egg+12]. Qi and Dai show that “viewers prefer a scenario in which a single but long freeze event occurs to a scenario in which frequent short freezes occur” [QD06]. Their other findings match those of Hossfeld et al. and Egger et al., they furthermore found out that a “one frame length distortion is unrecognizable for viewers” [QD06]. Krishnan and Sitaraman did research on the impact of start-up delays. They found out, that an “increase in startup [sic] delay causes more abandonment of viewers” [KS12]. Further, “viewers are less tolerant of startup [sic] delay for short videos in comparison to longer videos” [KS12]. “Viewers watching videos on a better connected computer or device have less patience for startup [sic] delay and so abandon sooner. [...] We can see that viewers abandon significantly less on mobile in

comparison with the other categories, for a given startup [sic] delay" [KS12]. To summarize, Hossfeld et al., Egger et al., Qi and Dai, and Krishnan and Sitaraman show that

- stalling during video playback has a very bad influence on the quality of experience while watching a video,
- one longer stalling/freeze event is preferred to more shorter ones by some users, others prefer more stalling events as a feedback mechanism, and
- most users prefer initial delays which should not be too long to avoid abandonment.

In annotated interactive non-linear videos, the additional information enhances the download volume of a scene and parallel storylines increase the download volume for future scenes. In addition, users may behave differently in varying video-structures having their origins in different usage scenarios (like e-learning or tours). Different end user devices (smart phone, ultra-book, PC) provide different cache and bandwidth capabilities which have to be taken into account. It also has to be taken into consideration that only parts of the video are needed at the end-user device, because not all scenes and annotations may be part of the video which is actually watched by the viewer. If the whole video is downloaded, an already limited download volume (for example on a smart phone) is wasted. These prerequisites lead to the following research questions:

What does the communication architecture of annotated interactive non-linear videos look like and how do single components interact in it?

How can a starting point for playback, which avoids interruptions, be calculated?

How can a start-up delays be minimized?

How can the elements be scheduled for download from the server to the client?

By which criteria are elements deleted from the cache?

In order to be able to find appropriate solutions for different end user devices and viewer behavior, we have built a modular player framework. The essential part is the player logic. It allows a download and cache management that is able to adapt to the viewer behavior and the underlying hardware. Different approaches for each of the tasks have been implemented and are exchangeable in order to deal with different end user devices. (This paragraph was taken and adapted from our previous work [MH12].)

6.1. Communication Architecture

Our communication architecture consists of a client and a server as shown in Figure 6.1. We assume, that the annotated interactive non-linear video is stored on the server, which has no logic implemented, but provides all the contents needed by the client, including a control file, the videos, and the annotation files. The whole logic is implemented in the client. The client downloads, caches, and plays the video. We implemented two caches, a download buffer and a player cache for storing data after the progressive download. Each cache is modeled as one connected storage space without any further subdivision (neither fixed nor variable). A fixed division of the cache, for example into two parts, one for frames and one for annotations, is not useful due to the strong variety in the composition of frames and annotations from scenario

to scenario (or video to video). Even a fixed division for a single annotated interactive non-linear video may be counterproductive because of a very uneven distribution of annotations in the different scenes. A variable division of the cache requires adaptation algorithms which determine the sizes of the caches depending on a given environment and assign the available cache size to the different caches. A separated administration, of for example frames and annotations or elements of varying priorities, may be possible then, but requires well adapted algorithms for deletion to avoid pauses during playback due to gaps in a sequence of frames. For these reasons, we chose to use one single cache for playback. The download buffer is only used to buffer not completely downloaded elements. The total cache size $B(t)$ is the player cache size $B^{player}(t)$ (as described in Section 5.2.1). The download buffer is used to collect all bits of a media object or a frame. After downloading one complete object, it is moved into the player cache.

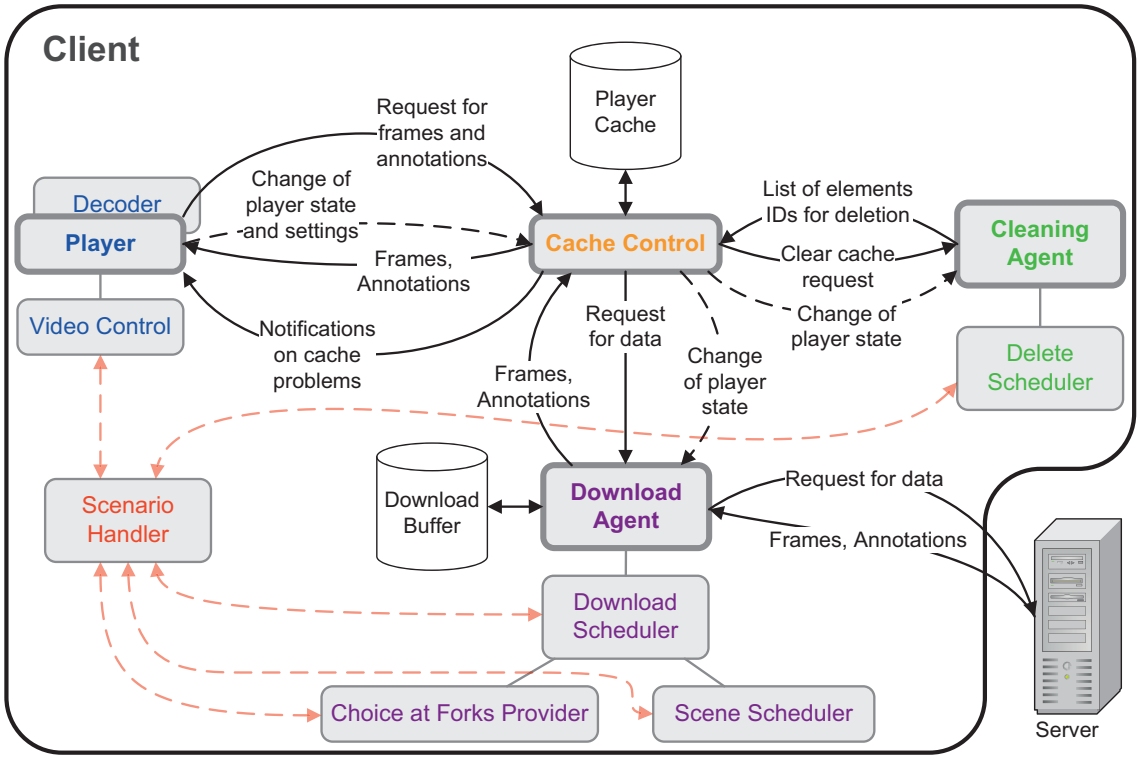


Figure 6.1.: Component Architecture and data flow of the video client.

The client knows which elements are needed at specific points in time (given by the control file) and requests them from the server. Our client consists of four instances where logic is implemented, namely player, download agent, cache control, and cleaning agent. All of them have to communicate with each other. An overview of the components and their interactions is given in Figure 6.1 and can be described as follows:

- The **download agent** manages all issues concerning the download of the video with the help of a ***download scheduler***, ***choice at forks provider***, and a ***scene scheduler***. The download scheduler determines the order for downloading the frames and annotations. The order depends on the scene schedule and the probabilities or relative frequencies (depending on the underlying conditions) for choosing a certain path at a fork. The download agent receives the elements from the server in the order given by the down-

load scheduler and stores them in the download buffer until an element is downloaded completely. Then the element is transferred to the player cache.

- The **player** is controlled by the ***video control*** which selects scenes and elements for display. The player receives the encoded frames and annotations from the player cache when they are needed for display in the video. The ***decoder*** is implemented in the player to decode videos and images for display. If there are any problems concerning the cache, the player is notified.
- The **cache control** recognizes changes of the player state or its settings and forwards them to the download agent or the cleaning agent if necessary.
- The **cleaning agent** creates a list of elements which have to be deleted from the cache with the help of a ***delete scheduler***. It is notified to select elements that can be deleted from the cache, if the filling level of the cache reaches a specified limit.
- The ***scenario handler*** provides calculated values which are needed by the other components. This avoids multiple calculations for a single value.

A UML sequence diagram is pictured in Figure 6.2. It shows the most important classes of our player framework: the player component which contains a decoder and a player, as well as classes for cache control, download, and delete. After all classes are initialized or cleared, the waiting for resources is started in the player and the download is started. When all resources needed for a certain scene are available, the playback is started. After a scene is finished, the follow-up scene is loaded. These steps are then repeated until the whole video is finished. A more extensive diagram containing the part shown in Figure 6.2 can be found in Appendix E, Figure E.1 on page 254.

The interactions between player, download, and cache are depicted more precisely in the UML state chart in Figure 6.3. Depending on the state of the player, the download, and the cache,

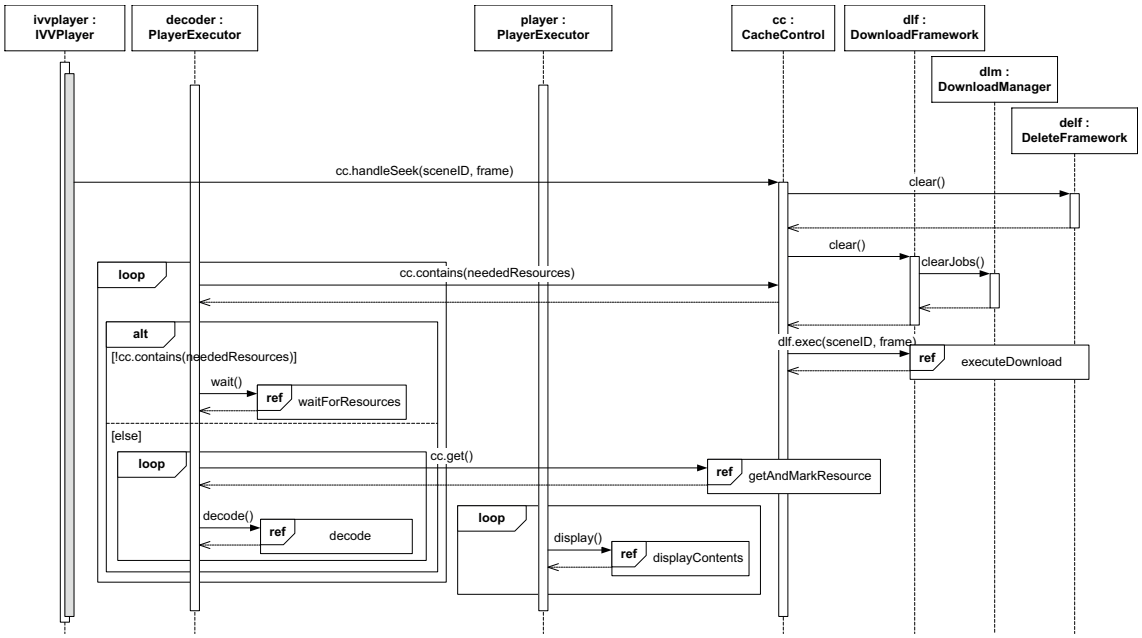


Figure 6.2.: Simplified UML sequence diagram illustrating the core of the simulation framework.

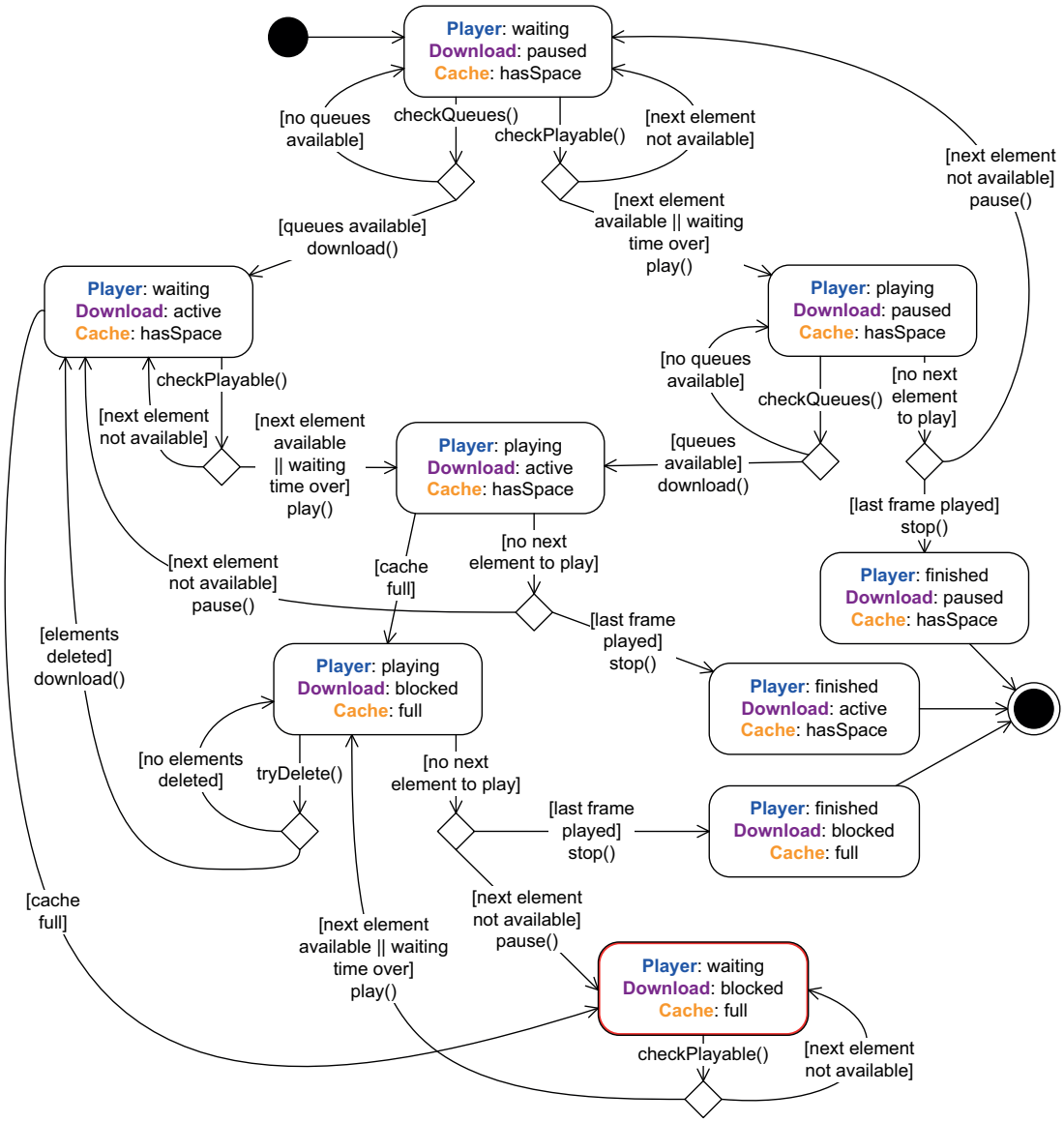


Figure 6.3.: UML state chart illustrating the connections between the state combinations of player, download, and cache. The red boxed state denotes the state in which the system deadlocks.

varying conditions have to be checked or actions have to be triggered. The framework starts with a waiting player, a paused download and an empty cache. Queues are created and the download becomes active active while the player is still waiting. Then, depending on the elements in the cache, the player may be in the states “playing”, “waiting”, or “finished”. The states of the download are “active”, “paused”, or “blocked”. The cache is either full or has space. A video can be finished in three different states.

The following sections of this work characterize each component more precisely. Thereby, usable algorithms and strategies are presented and described. (Parts of this section (6.1 Communication Architecture) were taken and adapted from our previous work [MH12].)

6.2. Global Calculations

Global calculations which are needed in more than one of the components are performed by the scenario handler. An important calculation is the determination of a frame of a scene that has to be downloaded into the player cache. This element denotes a point in time at which the scene can start playing while the rest of the data is downloaded in the background and the playback is not paused during the scene. We define some helper functions first and describe the calculation of the playback point later.

6.2.1. Sorting Elements of a Scene

To calculate the starting point for playback, first all elements have to be brought into an order and then the decisions on the starting point can be made. We define a partial order on the set of annotations of a scene $\mathcal{A}_{p_x}$. The partial order $\leq_a$ sorts all annotations according to their first time of display and their priority. Thereby, first all annotations are sorted according to their priority. Then, annotations of equal priority are sorted by means of their first time of display during the scene. Then all sublists are appended to each other with increasing Λ. The result is the tuple $\widehat{p}_x^a$. A basic algorithm called `SortSceneLinear`$(\widehat{p}_x^f, \widehat{p}_x^a, \Lambda)$ (see Algorithm 6.1) sorts all frames and annotations of one scene. It works in the following way: Both lists, $\widehat{p}_x^f$ and $\widehat{p}_x^a$ are combined to a single list. Doing that, the annotations are placed before the frame which they are displayed with during playback. If more than one annotation is displayed at a frame, they are inserted in a random order. As a result all elements are sorted in a linear order. Figure 6.4 shows two exemplary resulting lists (Figure 6.4, middle and bottom) of the sorting algorithm for a given display schedule where annotations a_1 and a_3 are displayed at the beginning of the scene, annotation a_4 is displayed at frame 60, a_5 is displayed at frame 100, a_2 is displayed at frame 170, and a_6 is displayed at frame 200 (Figure 6.4, top). The time where an annotation is hidden has no influence on the download schedule and therefore, is irrelevant. Assuming that all elements of a scene have the same priority, `SortSceneLinear`$(\widehat{p}_x^f, \widehat{p}_x^a, \Lambda)$ sorts the annotations before the frames they are displayed with (a_1 and a_3 before frame 1, a_4 before frame 60, and so on) (Figure 6.4, center). If we assume, that the same elements now have different priorities, then the audio and video annotations (a_4, a_5, and a_6 in our example) have a lower priority, for example. Then these elements are inserted at the end of the queue after the last frame. The annotations a_4, a_5, and a_6 may not be displayed during playback (Figure 6.4, bottom). The average runtime of `SortSceneLinear`$(\widehat{p}_x^f, \widehat{p}_x^a, \Lambda)$ is $O(n+m)$, where n is the number of frames ($n = dim(\widehat{p}_x^f)$) and m is the number of annotations ($m = dim(\widehat{p}_x^a)$) of the scene. The best and the worst case are also $O(n+m)$, because every element has to be added to the queue, no elements can be left out, and the queues are presorted.

6.2.2. Calculation of the Starting Point

The total duration for downloading a scene can be calculated as described in Function 6.1, assuming a linear download schedule of all elements of a scene. It is calculated by the sum of the sizes of the frames and annotations of a scene which is then divided by the transmission bandwidth BW. This sum can be split into three parts. The margin between the terms of the sum are the indices of frames f_m or f_n, $0 \leq m \leq n \leq dim(\widehat{p}_x^f)$, $m, n \in \mathbb{N}^+$. $P1$ is the part of

Input: list of frames $\widehat{p}_x^f$, list of annotations $\widehat{p}_x^a$, priority Λ
Output: list of elements $\widehat{p}_x^e$ according to the scheduling constraints

Initialization
`Clear(`$\widehat{p}_x^e$`)`

Enqueueing Scene
$j \leftarrow 0$
for $k \leftarrow 0$ **to** $dim(\widehat{p}_x^f)$ **do**
  if $(j < dim(\widehat{p}_x^a))$&&(`startFrame(`$a_j$`)` $== f_k$) && (`GetPriority(`a_j`)` $\leq \Lambda$) **then**
    $\widehat{p}_x^e$`.Append(`a_j`)`
    $j \leftarrow j + 1$
  end
  $\widehat{p}_x^e$`.Append(`f_k`)`
end
for j **to** $dim(\widehat{p}_x^a)$ **do**
  $\widehat{p}_x^e$`.Append(`a_j`)`
  $j \leftarrow j + 1$
end
return $\widehat{p}_x^e$

Algorithm 6.1: SortSceneLinear

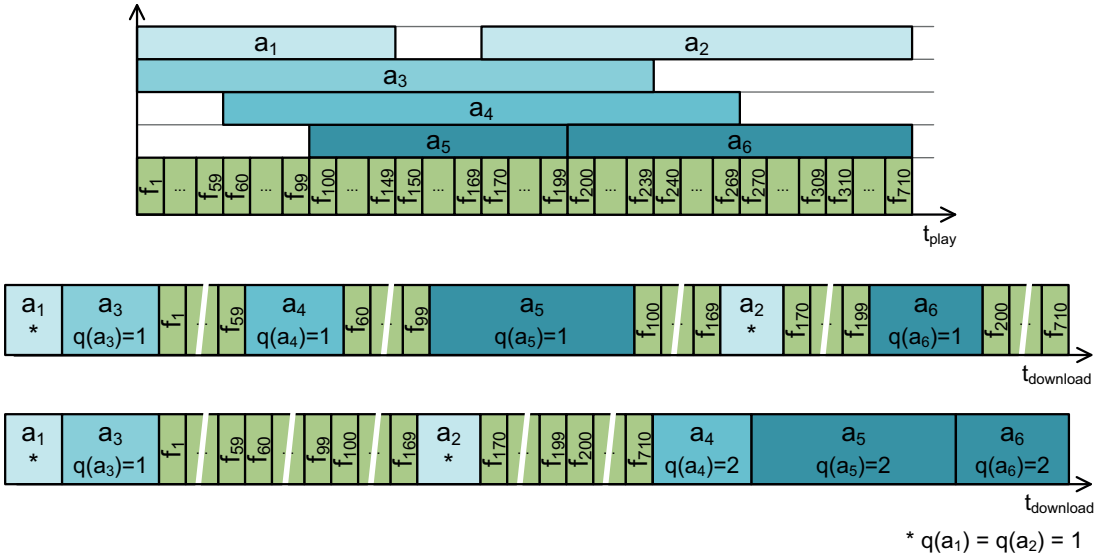


Figure 6.4.: A playback schedule of a scene (top) and linear download schedules assuming the same priority for all elements (center) as well as different priorities (bottom).

the video with annotation priorities below a certain level or if only equal priorities are used, which has to be downloaded for playing the video without reload[1], assuming that the scene can be played after frame f_{m-1} is downloaded to the client. $P2$ is the part of the video which has to be downloaded for playing the video and all its annotations without reload, assuming

[1]The phrase "reload" defines the behavior of a video player when it pauses and buffers/downloads more data needed for playback.

that the scene can be played after frame f_{n-1} is downloaded to the client. $P3$ is the part of the download that can be loaded after the scene is selected, if the download has the full available bandwidth. Therefor a reschedule in the download scheduler may be necessary because these elements have to be downloaded immediately after the scene was selected. Figure 6.5 shows the example download schedule from Figure 6.4 with possible parts P1, P2, and P3. Thereby, all elements up to frame 170 have to be downloaded to the client to be able to play the scene and high prioritized annotations without reloads. If all elements up to frame 710 are downloaded, it may be possible to play the whole scene without reloads (depending on the point in time where the annotations are displayed). While we always add the annotation before the frame, the annotations displayed at a particular frame are contained in the cache as well. We furthermore assume for our calculations, that the whole scene fits into the cache and each element is downloaded once for the scene.

$$dl : \mathcal{P}_{\mathcal{V}} \rightarrow \mathbb{N}^+, p_x \mapsto dl(p_x) \tag{6.1}$$

$$dl(p_x) := \frac{\sum_{i=0}^{dim(\widehat{p}_x^f)} s(f_i) + \sum_{j=0}^{dim(\widehat{p}_x^a)} s(a_j)}{BW}$$

$$= \underbrace{\frac{\sum_{i=0}^{m-1} s(f_i) + \sum_{j=0}^{k-1} s(a_j)}{BW}}_{P1} + \underbrace{\frac{\sum_{i=m}^{dim(\widehat{p}_x^f)} s(f_i) + \sum_{j=k}^{dim(\widehat{p}_x^a)} s(a_j)}{BW}}_{P2}$$

$$= \underbrace{\frac{\sum_{i=0}^{m-1} s(f_i) + \sum_{j=0}^{k-1} s(a_j)}{BW}}_{P1} + \underbrace{\frac{\sum_{i=m}^{n-1} s(f_i) + \sum_{j=k}^{o-1} s(a_j)}{BW}}_{P2} + \underbrace{\frac{\sum_{i=n}^{dim(\widehat{p}_x^f)} s(f_i) + \sum_{j=o}^{dim(\widehat{p}_x^a)} s(a_j)}{BW}}_{P3}$$

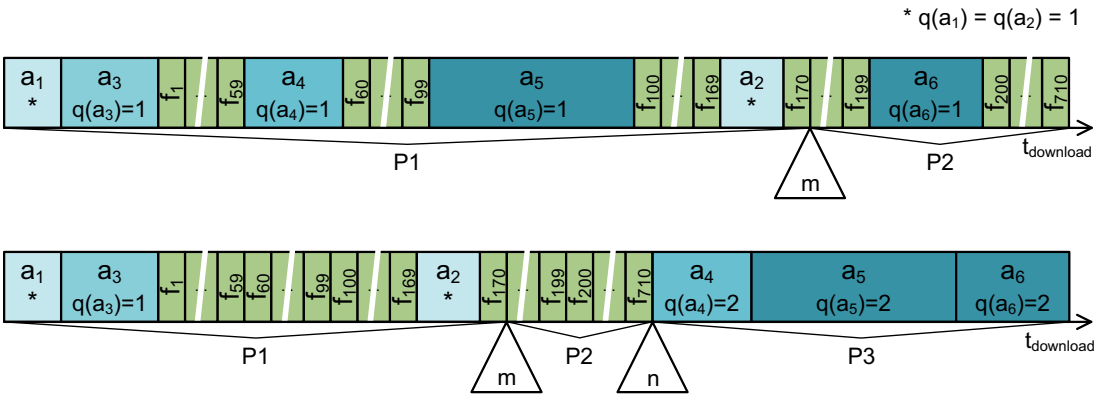


Figure 6.5.: Representation of $P1$, $P2$, $P3$, m, and n on a linear download schedule assuming different priorities.

An optimization problem is posed to calculate the best point in time (the PLAY_MIN_REL-point m or n) to start the playback for minimizing the number of reloads during a scene as formalized in Equation 6.2. The time needed for downloading the rest of the elements has to be smaller than the time for playing the video. These conditions can be formalized as an optimization problem as stated in Equations 6.2 to 6.6. The δ has to be kept as large as

possible, because the larger $dim(\widehat{p}_x^f) - m$ or $dim(\widehat{p}_x^f) - n$ (and thus δ) is, the smaller is m or n and the sooner starts the playback. Side condition 6.3 defines the criteria for playability of the video without reloads. It subtracts the download duration of all elements at the beginning of a scene up to frame m or n (including the annotations) (subtrahend) from the whole download duration of a scene (minuend). The difference has to be smaller than the playback time of the scene. This side condition 6.3 is only applicable, if the cache size is large enough. Constraints 6.4 to 6.6 define straight forward determining factors.

Maximize

$$\delta = dim(\widehat{p}_x^f) - m \text{ resp. } \delta = dim(\widehat{p}_x^f) - n \tag{6.2}$$

Constraints

$$dl(p_x) - \frac{\sum_{i=0}^{n} s(f_i) + \sum_{j=0}^{o} s(a_j)}{BW} < \quad dl(p_x) - \frac{\sum_{i=0}^{m} s(f_i) + \sum_{j=0}^{k} s(a_j)}{BW} < \frac{dim(f_1, \ldots, f_m)}{r} \tag{6.3}$$

$$0 \leq m \leq n \leq dim(\widehat{p}_x^f) \tag{6.4}$$

$$m, n, k, o > 0 \tag{6.5}$$

$$dl(p_x), s(f_i), s(a_k), dim(\widehat{p}_x^f), r, BW > 0 \tag{6.6}$$

The algorithm `GetStartFrame`$(\widehat{p}_x^f, \widehat{p}_x^a, \Lambda, r, BW)$ solves the optimization problem stated in Equations 6.2 to 6.6 and calculates the frame f_m from which on the video can be played without reloads. This calculation is described by Algorithm 6.2 and works the following way: First the algorithm calculates the download duration of a scene p_x, $dl(p_x)$, and creates a list of all elements of a scene $\widehat{p}_x^e$. Then the download duration of each element of the scene is subtracted from the whole duration for the download step by step. Doing this, each viewed element is appended to the list $\{f_1, \ldots, f_m\}$. When the element is inserted after which the remaining download duration is smaller than the playback duration, frame f_m has been found. The average runtime of `GetStartFrame`$(\widehat{p}_x^f, \widehat{p}_x^a, \Lambda, r, BW)$ is $O(3 \cdot (n+m))$, where n is the number of frames and m is the number of annotations of the scene. The sorting of the elements as well as the calculation of the download duration have an average and worst case runtime of $O(n+m)$, because each element has to be selected/added once. The calculation of the start frame has a best case runtime of $O(1)$ if the first element is selected and an average and worst case runtime of $O(n+m)$ if another frame is selected. The algorithm works optimal with regard to finding the best frame (smallest frame index), because of the two functions where one is strictly decreasing and the other is strictly increasing. Combined with a step-by-step iteration on the values inserting in the functions, the smallest frame index can be found. Depending on the last element before playability, the start point may be between the selected frame and the frame before that, but due to the fact of calculating with the sizes of whole frames and whole annotations, the latter one is chosen to make sure that the scene can be played without pauses. (This section (6.2 Global Calculations) was taken and adapted from our previous work [MH12].)

Input: list of frames $\widehat{p}_x^f$, list of annotations $\widehat{p}_x^a$, priority Λ, frame rate r, bandwidth c_{BW}
Output: frame f_m

Variables
tr is the remaining time
fc is a counter for frames
E is a list of elements
e_i is one element of E

Initialization
$fc \leftarrow 0$
$E \leftarrow$ `SortSceneLinear(`$\widehat{p}_x^f, \widehat{p}_x^a, \Lambda$`)`
$tr \leftarrow$ `CalcDownloadDuration(`E`)`

Get Frame
for $i \leftarrow 0$ **to** `Size(`E`)` **do**
  $tr \leftarrow tr - \frac{e_i.ElementSize}{c_{BW}}$
  if $e_i \in \widehat{p}_x^f$ **then**
    $fc \leftarrow fc + 1$
  end
  if $tr < \frac{fc}{r}$ **then**
    $f_m \leftarrow e_i.FrameNumber$
    return f_m
  end
end
return $dim(\widehat{p}_x^f)$

Algorithm 6.2: GetStartFrame

6.3. Video Playback/Start Time Strategies

To play annotated interactive non-linear videos, an XML control file as described in Section 3.1.2 is evaluated. The video control fetches encoded elements from the cache for display in the player. We assume that after decoding frames and annotations with an appropriate decoder, they are displayed without delay. Annotations are hidden after a timeout occurs.

If priorities are assigned to annotations, the player may start playback even if not all elements of the scene are in the cache. In this case, only those annotations need to be available in the cache, which have the priority levels that are needed for playback. Nevertheless, it is possible, that elements of lower priority levels are displayed during playback, if they are loaded into the cache, when cache size, download bandwidth, and download strategies are chosen accordingly.

The point in time at which the player starts the playback of a scene can be set differently. We have defined the following points for starting the playback of a scene in this work:

- **PLAY_SCENE**: The playback only starts when the whole scene is cached. This allows the viewer to jump forward and backward in a scene without pauses (if the cache is large enough).

- **PLAY_MIN_REL(f_m)**: The start of the playback is determined by the following optimization goal: minimize the number of pauses during playback of a scene. No pause is needed in the best case, where the $P1$ part (see Equation 6.1) of the scene can be downloaded into the cache. Priorities of elements are ignored thereby.
- **PLAY_MIN_REL_PRIO($f_{m/n}, \Lambda$)**: The start of the playback is determined by the following optimization goal: minimize the number of pauses during playback of a scene while only elements of a certain priority level are displayed using start frame f_m. No pause is needed at best. In this case, the $P1$ part (see Equation 6.1) of the scene can be downloaded into the cache. If the video is started at frame f_n, some of the lower prioritized annotations may be displayed, too. The playback of the parts $P1$ and $P2$ without pauses is possible if they can be downloaded into the cache.
- **PLAY_STARTUP(f_x)**: The playback starts when a given amount of frames up to frame f_x with $x \leq dim(p_i)$ (see Section 5.1.2) is cached and can be played without pauses. The further playback depends on the external limitations or video-dependent conditions, pauses may occur.

Using the PLAY_STARTUP strategy, a determined small waiting time at the beginning of a scene can be assured. But, if the transmission bandwidth is small, this strategy may lead to more pauses during the playback of a scene, which reduces the viewing experience. The PLAY_SCENE, the PLAY_MIN_REL, and the PLAY_MIN_REL_PRIO strategy avoid reloads during the playback of a scene, but lead to longer waiting times at the beginning of a scene, because the scene only starts playing, if a certain amount of data is cached. With small cache sizes, both strategies may also lead to pauses during a scene.

6.4. Download Scheduling

The download agent loads the elements from the server as provided by its input queues. We download annotations as a whole. Different download schedules are resulting from different algorithms and strategies as well as external and video-dependent conditions which we have to deal with.

6.4.1. Constraints

In order to make sure that the player will have the correct elements in the cache, the constraints stated in Definition 6.1 have to be satisfied when scheduling the download linearly. We therefore first define some download-related functions:

- The download deadline d_f of an element e_i is defined as $d_f : \mathcal{E}_\mathcal{V} \rightarrow \mathbb{N}^+, e_i \mapsto d_f(e_i)$ with $d_f(e_i) < d_f(e_j)\ \forall e_i < e_j,\ e_i, e_j \in \mathcal{E}_\mathcal{V}$
- The download duration d_d of an element e_i is defined as $d_d : \mathcal{E}_\mathcal{V} \rightarrow \mathbb{N}^+, e_i \mapsto d_d(e_i)$ with $d_d(e_i) < d_d(e_j)\ \forall s(e_i) < s(e_j),\ e_i, e_j \in \mathcal{E}_\mathcal{V}$
- The start time for the download d_s of an element e_i is defined as $d_s : \mathcal{E}_\mathcal{V} \rightarrow \mathbb{N}^+, e_i \mapsto d_s(e_i)$
- The display time d_t of an element e_i is defined as $d_t : \mathcal{E}_\mathcal{V} \rightarrow \mathbb{N}^+, e_i \mapsto d_t(e_i)$

Definition 6.1 (*Valid Schedule for one Scene $\mathcal{S}_{p_x}$*)

A linear schedule $\mathcal{S}_{p_x}$ for scene $p_x \in \mathcal{P}_\mathcal{V}$ is valid, if it complies with the following constraints.

- ***Playback constraint***
  $\forall e_i \in \mathcal{E}_{p_x} : d_f(e_i) < d_t(e_i)$*: the deadline $d_f(e_i)$ of the download has to be earlier than the display time $d_t(e_i)$ of the element.*
- ***Deadline constraint***
  $\forall e_i \in \mathcal{E}_{p_x} : d_s(e_i) + d_d(e_i) \leq d_f(e_i)$*: the download $d_d(e_i)$ has to be finished at the deadline $d_f(e_i)$ of an element, so the start time for the download $d_s(e_i)$ has to be smaller than $d_f(e_i) - d_d(e_i)$.*
- ***Order constraint***
  $\forall f_i, f_j \in \{\pi_1(\widehat{p}_x^f), \ldots, \pi_n(\widehat{p}_x^f)\}, i < j : d_f(f_i) < d_f(f_j)$*: the deadline d_f of frame f_i is set before the deadline d_f of frame f_j, because frame f_i is displayed before frame f_j.*
- ***Annotation constraint 1***
  $\forall (f_i, A_i) \in \{\pi_1(p_x), \ldots, \pi_n(p_x)\}, q(a_k) \leq \Lambda, a_k \in A_i : d_f(a_k) < d_f(f_i)$*: For all annotations with a given priority which is smaller than Λ, the deadline d_f of each of these annotations a_k must not be after the deadline d_f of the first frame f_i the annotation a_k is displayed with. At playback we assume that if a frame is in the cache, the associated annotations are in the cache, too.*
- ***Annotation constraint 2***
  $\forall a_k, a_o \in \mathcal{A}_{p_x}, q(a_k) < q(a_o) : d_f(a_k) < d_f(a_o)$*: the deadline d_f of a higher prioritized annotation a_k must not be after the deadline d_f of an annotation a_o with a lower priority.*

The playback constraint states that a frame has to be downloaded into the buffer before it is displayed by the player. The deadline constraint allows the calculation of the latest start time for a download by subtracting the download duration from the deadline at which the element has to be in the cache. Annotation and order constraints ensure the elements are downloaded in the right order. The earlier a frame is displayed, the earlier it is downloaded. Annotations are always downloaded before the frame they are displayed with, if they have a high enough priority. This ensures that they are in the cache when the frame they are displayed at is shown. Annotations with a lower priority are downloaded after those with a higher priority. This constraint avoids additional calculations in download and playback strategies because it ensures that high prioritized annotations are in the cache.

We schedule all elements in one single queue which is then downloaded at the full transmission bandwidth. It would also be possible to share the transmission bandwidth for parallel downloads, for example between frames and annotations or between two or more scenes which are downloaded due to pre-fetch strategies. Furthermore, it would be possible to use priorities to build different download queues. A usable algorithm is the “Predictive Deadline (PD) Scheduling algorithm” described by Miller [Mil92]. Thereby, “processes are required to submit estimates of execution times as well as a deadlines [sic] for operations they execute. In addition, each process must be assigned a static priority when it is created” [Mil92]. Using the PD scheduling in annotated interactive non-linear videos, the execution time equals the download time which can be calculated given the element size and the download bandwidth. Deadlines can be set by the download constraints described in Definition 6.1. Furthermore, it

would be possible to assign priorities to all annotations and use a fixed but high priority for the frames of a scene. The parallel processing of download queues may allow the application of more fine-grained download strategies while one single download queue may be easier to reschedule.

6.4.2. Pre-fetch Strategies

Pre-fetch strategies determine which elements of a scene are inserted into the download queue by the download scheduler. Both the order of the elements of a scene as well as the number of elements which should be inserted into the queue have to be defined. First, a sorted list of all frames and annotations is created by Algorithm 6.1. In the next step, it has to be decided what portion of the lists is "unlocked" for being inserted into the download queue. We introduce two pre-fetch strategies hereafter, one is called PREFETCH_SL, the other one PREFETCH_FF. The first one unlocks a certain amount of the download queue of a scene (frames and annotations, which meet the constraints described in Definition 6.1) and a certain depth from the current scene on for download. The second one unlocks a defined number of consecutive future frames (considering each path). The strategies can be described as follows:

- **PREFETCH_SL** The pre-fetch strategy PREFETCH_SL($z_{SL}, \gamma, \Lambda, dist$), $z_{SL} \in \mathbb{R}^+, z_{SL} \geq 1, \gamma \leq dim(p_x), dist \in \mathbb{N}^+, \Lambda \in \mathbb{N}^+$ takes parts of scenes and the distance between scenes into account. Therefor, four values have to be determined.

  The first three values define what part of the scene will be unlocked for download scheduling up to a given frame as stated in Function 6.7 (see also Figure 6.6). The whole scene will be unlocked for the queue, if the first option with $z_{SL} = 1$ is selected. Choosing the second option, the part of the scene is enabled for the queue which is needed for a playback of the scene without reloads with $z_{SL} = 1$. Choosing the third option, the part of the scene is enabled for the queue which is needed for a playback of the scene and a display of annotations with a certain priority higher than Λ without reloads with $z_{SL} = 1$. A specific percentage of the described amounts is downloaded with $z_{SL} \geq 1$. This may be useful if the cache is not full, bandwidth is available and scenes in the foreseeable future could be downloaded already.

$$b_1 : \widehat{p}_x^e \times \mathbb{R}^+ \times \mathbb{N}^+ \times \mathbb{N}^+ \rightarrow \widehat{p}_x^e, (\widehat{p}_x^e, z_{SL}, \gamma, \Lambda) \mapsto b_1(\widehat{p}_x^e, z_{SL}, \gamma, \Lambda) \tag{6.7}$$

$$b_1(\widehat{p}_x^e, z_{SL}, \gamma, \Lambda) := \begin{cases} \pi_{1,n}(\widehat{p}_x^e) \text{ with } n = \left\lceil \frac{dim(\widehat{p}_x^e)}{z_{SL}} \right\rceil, & \text{if } \gamma = dim(\widehat{p}_x^e), \Lambda = 1 \text{ (cf. whole)} \\ \pi_{1,n}(\widehat{p}_x^e) \text{ with } n = \left\lfloor \frac{m}{z_{SL}} \right\rfloor, & \text{if } \gamma = m, \Lambda = 1 \text{ (cf. playable)} \\ \pi_{1,n}(\widehat{p}_x^e) \text{ with } n = \left\lfloor \frac{m}{z_{SL}} \right\rfloor, & \text{if } \gamma = m, \Lambda > 1 \text{ (cf. playableprio)} \end{cases}$$

  The value $dist$ reveals the depth from the current scene on up to which the scenes are unlocked for download scheduling (see also Figure 6.6). After doing a breadth-first search on the scene graph resulting from the DFA from Definition 5.11, starting at the current scene $p_{current}$, the distance $dist_{BFS}$ (described as $d[u]$ in [Cor+04, p. 536]) between the current and the following scenes is known. Function 6.8 defines the distance from the current scene to one of the following scenes determined by the breadth first search. All scenes p_x having a smaller distance than the chosen value ($dist_{BFS}(p_{current}, p_x) \leq dist$) will be scheduled for pre-fetch in the download strategy.

$$dist_{BFS} : \mathcal{P}_\mathcal{V} \times \mathcal{P}_\mathcal{V} \rightarrow \mathbb{N}^+, (p_{current}, p_x) \mapsto dist_{BFS}(p_{current}, p_x) \tag{6.8}$$

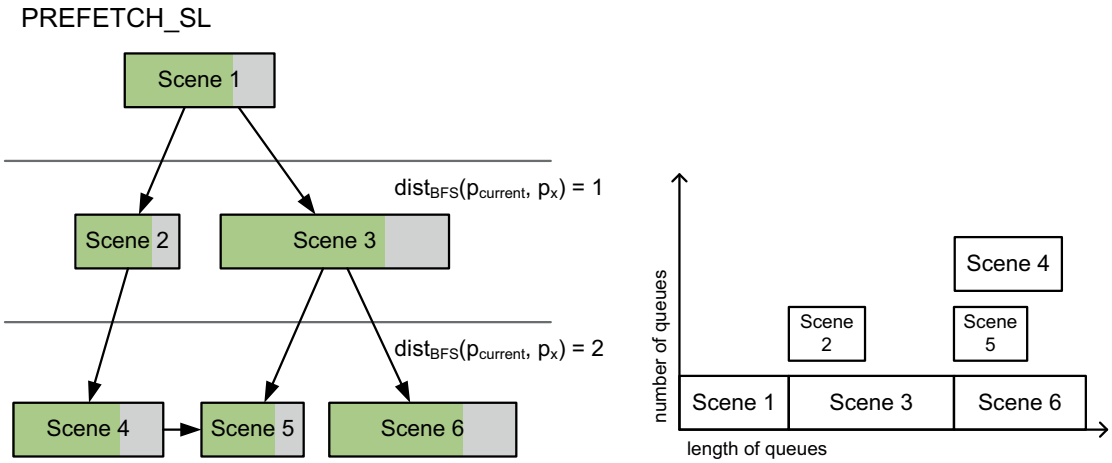


Figure 6.6.: Illustration of the pre-fetch strategy PREFETCH_SL with the resulting download queues.

- **PREFETCH_FF** The pre-fetch strategy PREFETCH_FF(z_{FF}), $z_{FF} \in \mathbb{N}^+$ takes a defined number of consecutive future frames whatever path is considered. From a start scene and start frame on, lists with consecutive z_{FF} elements are created for each possible path from the current frame of the current scene $p_{current}$ on.

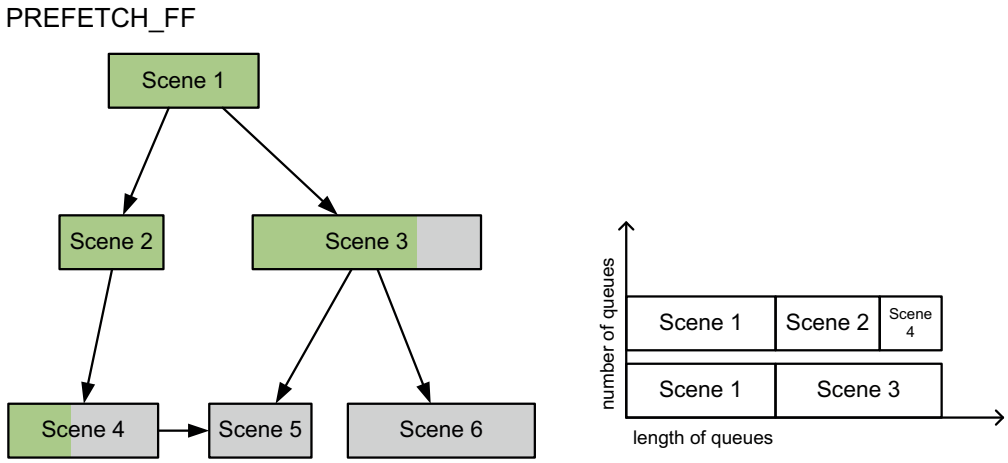


Figure 6.7.: Illustration of the pre-fetch strategy PREFETCH_FF with the resulting download queues.

Using this strategy, the resulting queues are not disjoint (in contrast to the PREFETCH_SL strategy). Elements that are in more than one queue are added to the download queue only once as the algorithm checks for elements which are already enqueued (see Subsection 6.4.4). As illustrated in Figures 6.6 and 6.7, are different download queues created by the described pre-fetch strategies. PREFETCH_SL loads more regular into the depth of the scene graph but does not provide a comprehensive sequence of elements, because scenes may not be downloaded completely. PREFETCH_FF loads a comprehensive sequence of elements independent from the depth of the scene graph. Some elements are added to the download queues more than once, the download scheduler only adds the first element to the download queue as a consequence.

6.4.3. Decision at Forks

At a fork in the video flow, user interactions need to be evaluated to decide which follow-up path is selected in the further course of the video. Furthermore, probabilities or relative frequencies have to be determined which indicate how important a path is for download. The modeling and evaluation of decisions can be accomplished with a huge variety of methods. Besides a button click as defined in Definition 5.11, a very simple method is the evaluation of Boolean expressions with a truth table. But more complex methods may be necessary in complex scenarios which allow the evaluation of other types of variables. On the one hand can decision trees as described by Forsyth ([For89, pp. 205-207], or Tello [Tel88, pp. 111-115]) be used to evaluate more complex expressions. On the other hand, techniques from soft computing like fuzzy systems, evolutionary computation, or artificial neural computing can be used to decide if a certain path should be selected based on perhaps imprecise or approximated values (for a basic description see [Cha08, pp. 2-10]).

While complex decision processes are possible at forks, as outlined previously, we assume that a single button is activated at a fork by the viewer in our test cases. The complexity of the decision process is irrelevant for the download and cache management described in this work. The only input needed for our algorithms is the relative frequency/probability of the selection of a certain path, because this is the only relevant information for our download scheduler.

We assume that a button and a following scene have a $1:1$ relationship. The scene graph described in Chapter 5, Figure 5.1 has only this kind of links. Thereby, the discrete feature of selecting one button has k characteristic features, where k is the number of buttons. The relative frequency $h_n(w_{i,j}), w_{i,j} \in \Sigma$ for clicking one of the k buttons can be derived from the behavior of n former viewers. The click behavior of former viewers of the video is logged. Each click is stored in a database. Thereby, the selected path and the hit/visits counter of the buttons are saved, as well as an anonymized viewer ID. The absolute frequency $H_n(w_{i,j})$ of the clicks is used to calculate the relative frequency $h_n(w_{i,j})$ with which a certain path in the model is chosen. Furthermore, the relative frequency of selecting a scene p_x is equal to the relative frequency of clicking the button $w_{i,x}$ linked with the scene, $h_n(p_x) = h_n(w_{i_1,x})$. Accordingly, the relative frequency $h_n(p_x)$ for selecting a scene can be calculated as stated in Equation 6.9, where p_x is the future scene, $p_{current}$ is the current scene, and the scenes $p_{current+1} \dots p_{x-1}$ are the scenes in between.

$$\begin{aligned} h_n(p_x) &= h_n(p_{current+1} \cap \ldots \cap p_x) \\ &= h_n(p_{current+1}) \cdot h_n(p_{current+2} | p_{current+1}) \cdot \ldots \cdot h_n(p_x | p_{current+1} \cap \ldots \cap p_{x-1}) \end{aligned} \tag{6.9}$$

When a new annotated interactive non-linear video is published on a web-server, the cold start problem for the relative frequencies has to be solved. This can be accomplished using a uniform distribution for each path at a fork. Another way to solve this problem is to let the author of the video define probabilities with which a fork may be selected by the viewer. Furthermore, it has to be decided how many viewers have to watch a video or visit a certain fork node until the values from the logs are used. After a specified number of viewers have watched the video, analyses on the scene graph are possible. While the strategies described in this work only deal with local decisions, it may be possible to apply more global strategies for download and cache management. Furthermore, not only optimizations are possible with the collected data, but based on the viewing behavior of single users, certain types of learners can also be identified. This enables the authors to improve their videos for different types

of learners. In the remainder of this work, we always use assigned probabilities instead of relative frequencies collected from real world scenarios.

6.4.4. Download Strategy

As we decided to use a serial download at the full available transmission bandwidth, instead of parallel downloads sharing the bandwidth, all elements selected by the pre-fetch strategy have to be brought into a linear order before they are loaded into the download queue and requested from the server.

We propose the algorithm `SortVideoLinear` for scheduling the download of scenes with equal distances to the current frame which relies on a modified "Efficient Fair Queuing Using Deficit Round Robin" from [SV95] as described in Algorithm 6.3 in an adapted way. It sorts the elements of paths at a fork into a linear order depending on the relative frequency $h_n(p_x)$ or the assigned probability with which a path/scene is selected by a viewer. Therefor, the value of the relative frequency or the probability is mapped to the quantum size for the scene. The elements of every scene are sorted linearly. All counters and the final queue are reset during the initialization phase (which is described as enqueuing module in [SV95]). If only one scene has to be enqueued, the queue is built and downloaded as it is. Parallel scenes with equal distance to the currently played frame are inserted into one list by their relative frequency to be played in the "enqueuing flow" part of Algorithm 6.3. Doing this, the algorithm processes the elements/scenes queue by queue for one distance. This part of the algorithm also decides if a Queue is ready for enqueuing depending on the used pre-fetch strategy. This is a modified version of dequeuing part of the algorithm described in [SV95]. A detailed example of the functioning of the original algorithm can also be found in [SV95].

Other algorithms may be usable which take the transmission bandwidth and the cache size into account. Therefore, our algorithms may be extended with existing algorithms and methods from Section 4.4. It may also provide benefits to use higher level algorithms which select fitting algorithms described in this work with appropriate parameters and react to changing transmission bandwidths or cache filling levels. A looking ahead download strategy which takes the cache filling level into account could be implemented this way. It could download more future elements if there is enough space in the cache. When the cache fills up more and more, it may be appropriate to decrease caching for the future. The evaluation of these ideas may be part of our future work. (This section (6.4 Download Scheduling) was taken and adapted from our previous work [MH12].)

6.5. Delete Strategies

Elements can be deleted from the cache if they are no longer needed. If the cache is full, selected elements have to be deleted. Several factors like the size of an element, possible future display, the last access on the element, and others can be taken into account to choose which element will be deleted. These factors can be weighted depending on the used delete strategy. Furthermore, the amount of data that has to be deleted to provide space for new elements when the cache is full has to be specified.

Input: list of lists of elements $((\widehat{p}_1^e),(\widehat{p}_2^e,\dots\widehat{p}_i^e),(\widehat{p}_j^e,\dots\widehat{p}_n^e),\dots) =: Scenes$
Output: sorted list of frames and annotations for download M

Variables
A is the list of active scenes
D_i is the queue of the scene p_i currently worked with
DC_i is the deficit counter for scene p_i
Q_i is the quantum size for scene p_i

Initialization
for $i \leftarrow 0$ **to** `Count(`*Scenes*`)` **do**
  $A = (\widehat{p}_1^e)$
  `Clear(`D_i`)`
  $DC_i = 0$
end
`Clear(`M`)`

Enqueueing Flow
while !`Empty(`*Scenes*`)` **do**
  while !`Empty(`A`)` **do**
    if `QueueReady(`D_i`)` **then**
      Queue $D_i \leftarrow$`FirstElement(`A`)`
      $DC_i = Q_i + DC_i$
      while ($DC_i > 0$ && !`Empty(`D_i`)` **do**
        $i.ElementSize \leftarrow$ `Size(FirstElement(`D_i`))`
        if $i.ElementSize \leq DC_i$ **then**
          if !M`.Contains(FirstElement(`D_i`))` **then**
            M`.Append(FirstElement(`D_i`))`
          $DC_i = DC_i - i.ElementSize$
        else
          `break`
        end
      end
    if `Empty(`D_i`)` **then**
      $DC_i = 0$
    else
      A`.Append(`D_i`)`
    end
  end
  A`.AppendNextList ()`
end

Algorithm 6.3: SortVideoLinear

6.5.1. Categories of Elements

Elements in the cache can be divided into four different categories. These may be treated differently when they are checked for deletion. The categories are described as follows:

- **Category A**: The element is unreachable at the current state.
- **Category B**: The element can be reached at rewind.
- **Category C**: The element is a future element of the currently played scene.
- **Category D**: The element can be displayed as part of the further course of the video.

With varying allowed user behaviors, one element may be assigned differently to one of the categories considering the same point in time during playback. We now illustrate these circumstances with the example graph from Figure 6.8, where the viewer watches the middle of "Scene 6" (orange triangle in Figure 6.8).

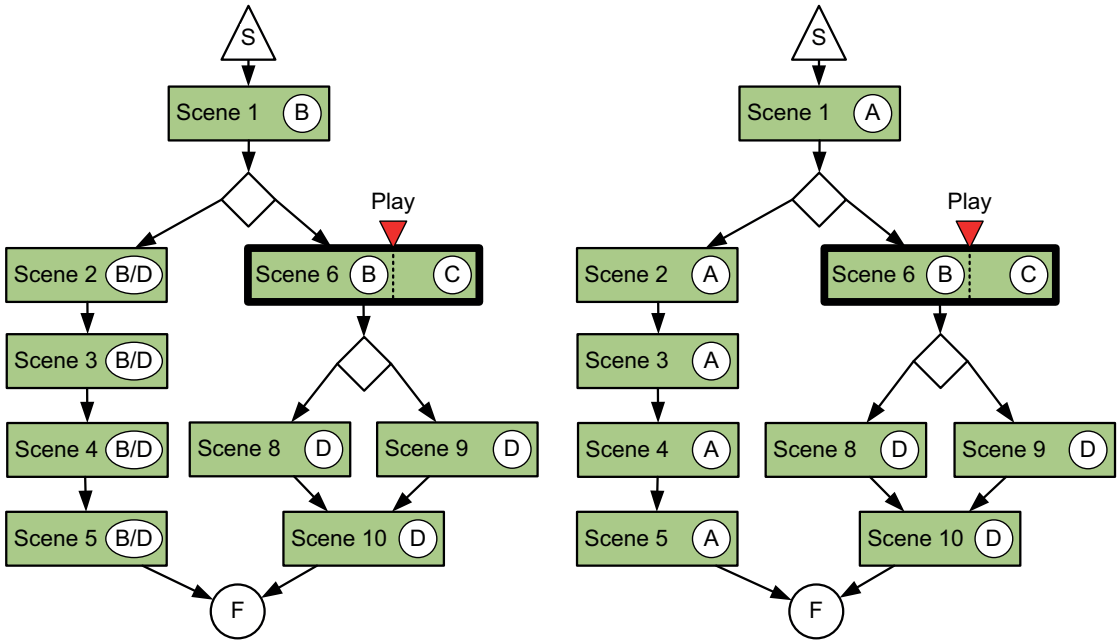


Figure 6.8.: Assignment of delete categories for different allowed user behaviors: non-restrictive (left) vs. restrictive (right).

Assuming a non-restrictive user behavior, that allows the user to rewind as far as wanted and allows to select a new path after rewinding beyond a fork, all elements may be always playable (see Figure 6.8, left). In this case, no element is assigned to category A. Elements that are played for sure if the user does not rewind and reselect are those of scene 6, category C and scene 10 in Figure 6.8 (left). Elements that can be reached at rewind and possibly a reselection (category B) are marked with an encircled B in Figure 6.8 (left). Those elements that may be played with reselection (mixture of categories B and D), are marked with an encircled B/D. Elements that can be displayed as part of the further course of the video (category D) are marked with an encircled D in Figure 6.8 (left). Assuming a restrictive user behavior, the user is only allowed to rewind in the currently played scene, no backward navigation is possible. In this case, the left path (scenes 2-5) as well as the first scene are in category A. Elements that

can be reached at rewind (category B) are marked with an encircled B in Figure 6.8 (right). Elements that are played for sure if the user does not rewind and reselect are marked with an encircled C in Figure 6.8 (right). Elements that can be displayed as part of the further course of the video (category D) are marked with an encircled D in Figure 6.8 (right). Other types of user behavior are possible. As the example shows, especially past (already played) scenes are categorized differently depending on the allowed user behavior.

6.5.2. Indices for Deletion

Our approach is to calculate an index value for each of the elements in the cache. Different indices can be defined for clearing the cache. In this work, we propose combinations of size function, distance function, and the relative frequency of choosing a path. After assigning the index to all elements in the cache, they can be sorted by value. The size function is defined in Section 6.2, Function 5.5 and returns the consumption of cache space for each element. The relative frequency of an element is calculated from the log of scene selections by former users as described in Section 6.4.3. The temporal distance between the current point in time and the point in time an element was hidden, is calculated by the distance function Δ shown in Function 6.10. It returns a number of frames. If an element was visited more than once or will be visited more than once, the smaller of the distances of the different visits $min(\Delta(e_i), \Delta(e_j))\ \forall\Delta(e_i),\ \forall\Delta(e_j)$ is used.

$$\Delta : \mathcal{E}_\mathcal{V} \rightarrow \mathbb{N}, e_i \mapsto \Delta(e_i) \tag{6.10}$$

$$\Delta(e_i) := \begin{cases} \infty, & e_i \text{ is unreachable} \\ -dim(f_{current}, \ldots, f_x), & e_i \text{ is a future frame } f_x \\ -dim(f_{current}, \ldots, f_x), & e_i \text{ is a future annotation first displayed at } f_x \\ dim(f_{current}, \ldots, f_x), & e_i \text{ is a past frame} \\ dim(f_{current}, \ldots, f_x), & e_i \text{ is a past annotation last displayed at } f_x \end{cases}$$

An index for deletion builds a partial order on the elements by the values assigned. This partial order can be described as follows: $(e_i, \ldots, e_k, \ldots, e_n)$, $g_j(e_i) < g_j(e_k)$ with the functions for g defined hereafter. The greater the value, the earlier an element is deleted. We defined four different indices, namely DELETE_SD, DELETE_LRU, DELETE_D_PROB, and DELETE_PRIO.

- **DELETE_SD(μ)**: DELETE_SD(μ) takes the size of an element $s(e_i)$ and the distance function $\Delta(e_i)$ into account. ∞ is assigned to unreachable elements, so they are sorted to the upper end of the index-list. A combination of the size of an element and the distance to the currently played element is assigned to all elements displayed already. Thereby, the terms of the sum can be weighted by $\mu \in [0..1]$. $-\infty$ is assigned to all elements in the future. The index function $g_1(e_i, \mu)$ of DELETE_SD(μ) is defined in Function 6.11.

$$g_1 : \mathcal{E}_\mathcal{V} \times [0..1] \rightarrow \mathbb{R} \cup \{-\infty, \infty\}, e_i \mapsto g_1(e_i, \mu) \tag{6.11}$$

$$g_1(e_i, \mu) := \begin{cases} \infty, & e_i \text{ in category A} \\ \mu \cdot \Delta(e_i) + (1 - \mu) \cdot s(e_i), & e_i \text{ in category B} \\ -\infty, & e_i \text{ in category C and category D} \end{cases}$$

- **DELETE_LRU**: The DELETE_LRU strategy only takes the distance function $\Delta(e_i)$ into account. It is an alteration from the well known LRU cache management strategy. The index function $g_2(e_i)$ of DELETE_LRU can be found in Function 6.12.

$$g_2 : \mathcal{E}_\mathcal{V} \to \mathbb{R} \cup \{-\infty, \infty\}, e_i \mapsto g_2(e_i) \tag{6.12}$$

$$g_2(e_i) := \begin{cases} \infty, & e_i \text{ in category A or category D} \\ \Delta(e_i), & e_i \text{ in category B} \\ -\infty, & e_i \text{ in category C} \end{cases}$$

- **DELETE_D_PROB**: The distance function Δ and the relative frequency h_n of an element which may be reachable from more than one scene in the scene graph are combined for the DELETE_D_PROB strategy. Function 6.13 defines the index function $g_3(e_i)$ for the DELETE_D_PROB strategy.

$$g_3 : \mathcal{E}_\mathcal{V} \to \mathbb{R} \cup \{-\infty, \infty\}, e_i \mapsto g_3(e_i) \tag{6.13}$$

$$g_3(e_i) := \begin{cases} \infty, & e_i \text{ in category A} \\ \Delta(e_i), & e_i \text{ in category B} \\ -\infty, & e_i \text{ in category C} \\ \Sigma h_n(p_x) \cdot \frac{1}{\eta}, & \eta = min\{\Delta(e_i) | e_i \in p_x \wedge e_i \text{ in category D}\} \end{cases}$$

- **DELETE_PRIO(Λ)**: DELETE_PRIO(Λ) takes the priority Λ of an element $q(e_i)$ and the distance function $\Delta(e_i)$ into account. ∞ is assigned to unreachable elements, so they are sorted to the upper end of the index-list. The index function $g_4(e_i, \Lambda)$ of DELETE_PRIO(Λ) is defined in Function 6.14.

$$g_4 : \mathcal{E}_\mathcal{V} \times \mathbb{N}^+ \to \mathbb{R} \cup \{-\infty, \infty\}, e_i \mapsto g_4(e_i, \Lambda) \tag{6.14}$$

$$g_4(e_i, \Lambda) := \begin{cases} \infty, & e_i \text{ in category A} \\ \Delta(e_i) \cdot q(e_i), & e_i \text{ in category B} \wedge q(e_i) \geq \Lambda \\ \Delta(e_i), & e_i \text{ in category B} \wedge q(e_i) < \Lambda \\ -\infty, & e_i \text{ in category C and category D} \end{cases}$$

The algorithm for the calculation of the index works as follows: First a breadth-first search on the cached elements is performed from the current scene into the future, as far as elements are there in the cache. All reached elements of the currently played scene are marked with the index value given for category C in the selected strategy, all other reached elements are marked with the index value given for category D in the selected strategy. Afterwards, all elements that are unmarked by the player and thus unreachable are classified as category A and marked with the index value for this category in the selected strategy. After that only elements of category B are remaining. Finally, the index value is calculated and assigned, if no value of category D is assigned yet. So an index is assigned to all elements in the player cache and the elements are sorted in a partial order.

We now illustrate the delete indices with an example. We assume that the viewer is not able to select a new path after jumping backward to a fork. In Figure 6.9 is scene 3 unreachable when the viewer watches scene 2. Scenes 4 and 5 are future scenes and scene 1 can be reached by jumping backwards. Different values are assigned to the elements in the cache depending on the used index, see Table 6.1. Scene 3 is unreachable and thus all elements have the value ∞. Elements from scenes 4 and 5 have ∞ or $-\infty$ depending on the index. Elements from scene 1 and 2 have values according to the functions defined for category B.

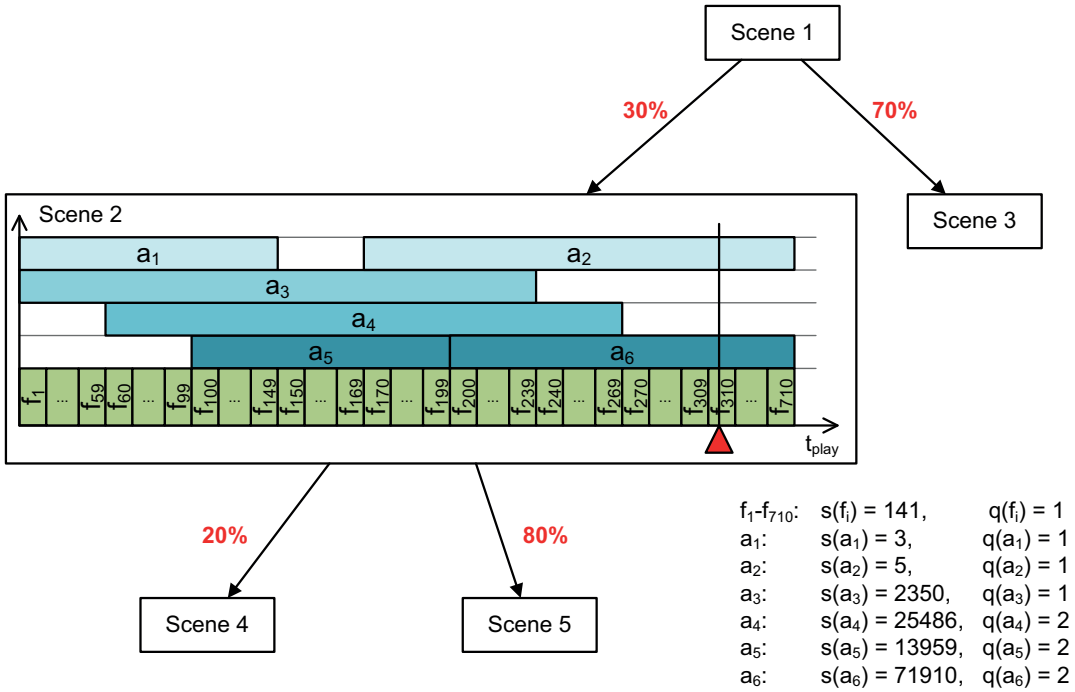


Figure 6.9.: Example graph for the calculation of delete indices.

6.5.3. Delete Threshold

With a given amount of space that is needed to be deleted from the cache, as many of the elements with the biggest values for g_1 to g_4 can be selected for deletion. This amount of data is specified by a percent value $perc$. As many elements are cleared from the cache until this percentage of the cache is empty. If the space in the cache that was emptied is not enough, the values are decreased gradually in order to allow the deletion of more elements and to avoid deadlocks. (This section (6.5 Delete Strategies) was taken and adapted from our previous work [MH12].)

6.6. Deadlocks

The strategies from the areas download, playback, and delete are self-contained. They operate independent of one another, but exchange messages about their current states (as described in Section 6.1). This approach requires approaches to avoid deadlocks for certain combinations of strategies, cache sizes, and transmission bandwidths. Especially settings with small caches are at risk of deadlocks. A deadlock occurs according to Coffman, Elphick, and Shoshani [CES71], if the following four conditions are met:

1. "***Mutual exclusion*** condition: tasks claim exclusive control of the resources they require.
2. ***Wait for*** condition: tasks hold resources already allocated to them while waiting for additional resources.

	DELETE_SD	DELETE_LRU	DELETE_D_PROB	DELETE_PRIO
Scene 1	$> 160 + s(e_i)$	> 310	> 310	> 310
Scene 2 - f_1	225	309	309	309
Scene 2 - a_1	156	309	309	309
Scene 2 - a_3	1329,5	309	309	309
Scene 2 - f_{59}	196	251	251	251
Scene 2 - f_{60}	195,5	250	250	250
Scene 2 - a_4	12868	250	250	500
Scene 2 - f_{99}	176	211	211	211
Scene 2 - f_{100}	175,5	210	210	210
Scene 2 - a_5	7084,5	210	210	420
Scene 2 - f_{169}	141	141	141	141
Scene 2 - f_{170}	140,5	140	140	140
Scene 2 - a_2	72,5	140	140	140
Scene 2 - f_{199}	126	111	111	111
Scene 2 - f_{200}	125,5	110	110	110
Scene 2 - a_6	36010	110	110	220
Scene 2 - f_{309}	71	1	1	1
Scene 2 - f_{310}	0	0	0	0
Scene 2 - f_{311}	$-\infty$	$-\infty$	$-\infty$	$-\infty$
Scene 2 - f_{710}	$-\infty$	$-\infty$	$-\infty$	$-\infty$
Scene 3	∞	∞	∞	∞
Scene 4	$-\infty$	∞	$-0,2 \cdot \frac{1}{\gamma}$	$-\infty$
Scene 5	$-\infty$	∞	$-0,8 \cdot \frac{1}{\gamma}$	$-\infty$

Table 6.1.: Calculated values for the proposed delete indices.

3. ***No preemption*** condition: resources cannot be forcibly removed from the tasks holding them until the resources are used to completion.

4. ***Circular wait*** condition: a circular chain of tasks exists, such that each task holds one or more resources that are being requested by the next task in the chain." [CES71]

A deadlock occurs if the cache is filled up to a certain level, no more elements can be deleted from category A, category B, and category D, and the frame from where the playback should start cannot be loaded into the cache. The conditions defined by Coffman, Elphick, and Shoshani in [CES71] are met the following way:

1. ***Mutual exclusion*** condition: The cache size B is limited to a certain amount of space and cannot be overcrowded.

2. ***Wait for*** condition: The playback task locks elements from the currently played scene for deletion, which have not been displayed yet. It in addition waits for a future frame to be able to start playback (according to the defined starting point for playback).

3. ***No preemption*** condition: Not displayed elements from the current scene cannot be removed from the cache until they were displayed in the player. The delete task cannot delete elements as a consequence.

4. ***Circular wait*** condition: The playback task blocks elements for deletion and occupies the cache thereby. The download task cannot load frames into the cache to reach the starting point for playback. The delete task cannot remove elements from the cache (see Figure 6.10)

Figure 6.10 shows an example where a deadlock occurs (see also red boxed state in Figure 6.3 in Section 6.1). The elements up to frame F169 can be loaded into the cache. The playback task waits for frame F200 to start playback. The delete task cannot delete any elements from the cache, because all of them are locked by the playback task. Thus the download task cannot load the missing elements into the cache, because no space is available.

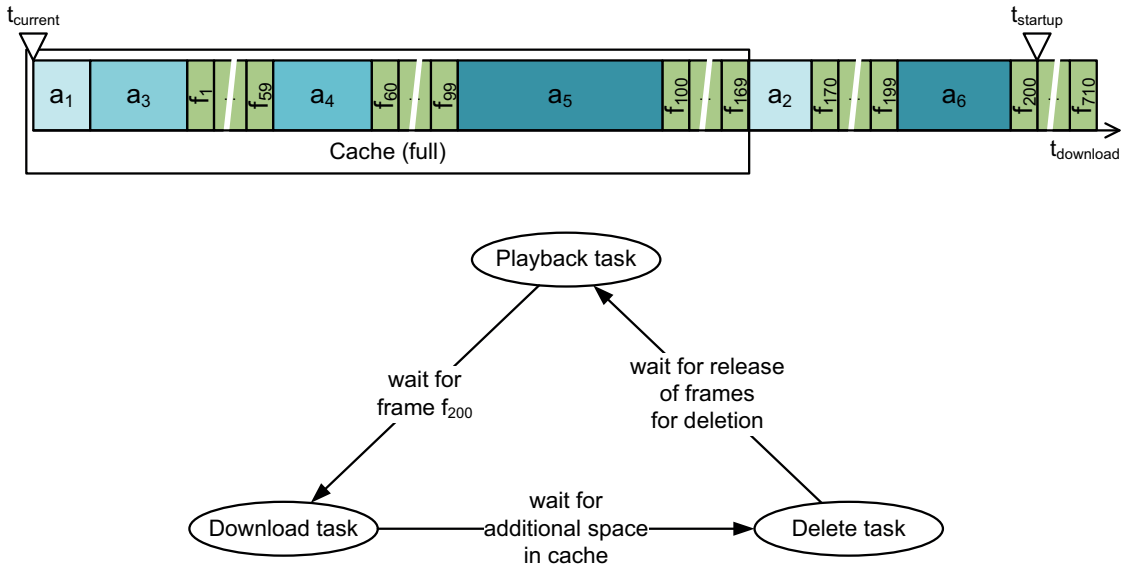


Figure 6.10.: A situation where a deadlock occurs: the playback cannot start, no elements can be deleted, and no additional elements can be loaded.

The described deadlock requires the player framework to start playback whether or not there are enough frames and annotations in the cache for playback. This usually leads to one or more pauses during a scene. It may furthermore be necessary to delete more elements as it is supposed to by the delete strategy and the defined amount that should be deleted (emergency delete). In this case even elements from category D or in the worst case category C will be deleted. This behavior may lead to increasing download volumes, but usually has no influence on the playback during a scene. It may result in longer waiting time at the beginning of the follow-up scenes of the scene in which the emergency delete is executed.

6.7. Summary

In this section, we described the download and cache management algorithms and strategies for annotated interactive non-linear videos. We first proposed a **communication architecture**. It consists of a download agent which controls the whole download with the help of a download scheduler, a choice at forks provider, and a scene scheduler. A player, consisting of a decoder and a video control element, displays all media contents. The cache control is a com-

munication unit which is used to send messages to other agents. The cleaning agent schedules elements for deletion. **Global calculations** are made by the scenario handler which provides the results to the other components of the architecture. Global calculations and algorithms are described, like the sorting of elements of a scene or the calculation of the starting point for playback. These are mainly performed by the scenario handler. In the **video play-pack** part of this section, we defined several points in time for start-up at the beginning of a scene. The **download scheduling** consists of three major subtasks. A scene scheduler implements pre-fetch strategies and ensures the compliance of scheduling constraints for single elements. Probabilities or relative frequencies are assigned to each choice at a fork which then have influence on the amount of data that are downloaded for a path. The download scheduler then creates and downloads queues depending on the probabilities at the forks and on the pre-fetch strategies. When the cache is full and additional space is needed, elements need to be removed by the delete scheduler using **delete strategies**. A delete strategy first assigns categories to the elements in the cache, then assigns values according to a selected delete index to each of the elements of each category, and then deletes a certain amount of data given by the threshold. A **deadlock** may occur when only a small cache is available, which then has to be resolved.

7. Evaluation

The algorithms and strategies described so far have to be tested for their usability with different bandwidths, cache sizes, and video structures. We want to show that the reloading times during a scene and the waiting times at the beginning of a scene are shorter using our download and cache management than using a traditional linking and loading behavior. The evaluation process has to take into account, that the suitability of a combination of strategies may depend on the underlying scene graph. All strategies furthermore need to be evaluated in comparison with each other. Metrics need to be selected from existing ones or defined from scratch which allow us to make statements about the suitability and quality of our strategies for download and cache management of annotated interactive non-linear videos. While the implementation of the player framework and its algorithms and strategies in different platforms is an extensive job which furthermore limits a direct comparison of the algorithms and strategies due to underlying hardware constraints, we decided to use a simulation framework for testing the algorithms and strategies. Furthermore, test cases are needed to examine the algorithms and strategies for their usability with different combinations of variables. We use five smaller test patterns to show that our algorithms and strategies are suitable for annotated interactive non-linear videos. These patterns are derived from the patterns in hypertext documents as described by Bernstein [Ber98]. Section 7.2.1 provides a detailed description of the patterns. In addition, we test the algorithms and strategies with a user generated real world scenario which combines some of the patterns.

After defining appropriate metrics and finding usable test patterns and scenarios, we want to answer the following questions with our evaluation:

> *Are the results of the simulations significant enough to derive statements from only one run of each test in a simulation?*
>
> *Which combination of algorithms/strategies results in the smallest number of frames to wait before playback averaged over all patterns?*
>
> *Which combination of algorithms/strategies results in the smallest waiting times before playback averaged over all patterns?*
>
> *Which combination of algorithms/strategies results in the fewest pauses during playback averaged over all patterns?*
>
> *Which combination of algorithms/strategies results in the smallest data volume of not watched elements averaged over all patterns?*
>
> *Which combination of algorithms/strategies results in the smallest data volume of repeatedly downloaded elements averaged over all patterns?*
>
> *Which combination of algorithms/strategies results in the smallest download volume averaged over all patterns?*

How do selected combinations of algorithms/strategies perform in more extreme settings (more annotations, wider patterns, disadvantageous path probabilities)?

Do the selected combinations of algorithms/strategies also show corresponding results in real world scenarios compared to the results from the patterns?

7.1. Performance Metrics

While watching annotated interactive non-linear video, two criteria are important. First, the playback of the video and the display of annotations has to be as fluent and with as few breaks as possible to provide an appealing viewing experience. Second, if the video is played on an end-user device with limited bandwidth volume settings, the amount of elements which are downloaded but not displayed should be as small as possible while providing an acceptable viewing experience. Several metrics can be used to evaluate the performance of download and cache management strategies for annotated interactive non-linear videos. Traditional metrics like those used in web proxy cache management (see [Won06]) could be applied, but are not meaningful enough to analyze the viewing quality or data volumes. Metrics like the hit-rate or the byte hit-rate ([Arl+00; Bah+02; BC08], and others), the memory hit-rate or disk hit-rate [ADM06], or the page hit-rate [GAGM09] indicate if an element (in our case a video or an annotation) is in the cache when it is requested, but do not provide any information about the accumulating time the viewer has to wait or what amount of data was downloaded but not watched. The number of replacements [GAGM09] and the delay-savings-ratio [Bah+02] are not applicable either because of the same reasons. The quality of experience of a set of strategies for download, cache, and delete management can either be measured on a small number of breaks in the video flow or in a minimization of the download volume while having an acceptable viewing experience. A further criterium is a short initial waiting time. The first and the last metric are described by ParandehGheibi et al. [Par+11] for linear streaming applications. They are described by our metrics WT_{start} and P_{sum}. The minimization of the download volume is not relevant for linear videos, because in these, all frames have to be downloaded. To describe and evaluate a possible download overhead, we use the metrics $DL_{not\ watched}$ and $RDLV$, which indicate the data volume of elements which are downloaded but not watched and the data volume of repeatedly downloaded elements. We furthermore take a look at the number of frames to wait at the beginning of a scene WF_{start} and the overall download volume DLV.

To define the metrics precisely, firstly we need to define some sets and variables. These are the tuple of elements of a path $\mathcal{P}ath_{\mathcal{V}}$, the set of downloaded elements $\mathcal{DX}_{\mathcal{V}}$, the set of downloaded but not watched elements $\mathcal{DXN}_{\mathcal{V}}$, and the set of repeatedly downloaded elements $\mathcal{DXR}_{\mathcal{V}}$:

- Tuple of elements of a path ($\mathcal{P}ath_{\mathcal{V}}$)
  The tuple of scenes of a path selected by the viewer or triggered by sequential scene changes is defined in Equation 7.1.

$$\mathcal{P}ath_{\mathcal{V}} := (p_1, \ldots, p_x), p_i \text{ watched by viewer}, 1 \leq i \leq x \tag{7.1}$$

- Set of downloaded elements ($\mathcal{DX}_\mathcal{V}$)
  The set of elements which are downloaded into the player cache is defined in Equation 7.2.

$$\mathcal{DX}_\mathcal{V} := \{(e_i, x_i) \mid e_i \text{ downloaded into the player cache as } i\text{th element}\} \subseteq \mathcal{E}_\mathcal{V} \times \mathbb{N}^+ \tag{7.2}$$

- Set of downloaded but not watched elements ($\mathcal{DXN}_\mathcal{V}$)
  The set of elements which were downloaded into the player cache but not watched by the viewer is defined in Equation 7.3.

$$\mathcal{DXN}_\mathcal{V} := \{(e_i, x_i) \mid (e_i, x_i) \in \mathcal{DX}_\mathcal{V} \wedge e_i \text{ not watched by the viewer}\} \subseteq \mathcal{DX}_\mathcal{V} \tag{7.3}$$

- Set of repeatedly downloaded elements ($\mathcal{DXR}_\mathcal{V}$)
  The set of elements which are repeatedly downloaded into the player cache is defined as in Equation 7.4.

$$\begin{aligned}\mathcal{DXR}_\mathcal{V} &:= \{(e_2, x_2) \mid (e_2, x_2) \in \mathcal{DX}_\mathcal{V} \wedge \exists x_1, x_2 : \pi_1(e_1, x_1) = \pi_1(e_2, x_2), x_1 \neq x_2\} \\ &= \{(e_i, x_i) \mid (e_i, x_i) \in \mathcal{DX}_\mathcal{V} \wedge e_i \text{ downloaded two or more times}\} \subseteq \mathcal{DX}_\mathcal{V}\end{aligned} \tag{7.4}$$

With the previously defined sets and points in time, metrics can be defined to describe the viewing experience and the quantity of downloaded data:

- WF_{start}: The calculated/determined frame number of the start frame according to the selected point in time where the playback could start without causing pauses during playback, summed up over all scenes in the path selected by the viewer as defined in Equation 7.5 with $dim(\mathcal{P}ath_\mathcal{V}) = x$. The elements in the cache are not considered for this metric.

$$WF_{start} := \sum_{i=1}^{x} x_i,\ x_i \text{ is the frame index of the determined start frame } f_{i,x_i} \text{ for scene } p_i \tag{7.5}$$

- WT_{start}: The amount of waiting time of the player summed up over all scenes while downloading data before the playback of a scene is started as defined in Equation 7.6. While the WF_{start} is based on a frame number which is determined independently from the elements in the cache, the result for WT_{start} may differ from that result, because already cached elements may shorten the waiting time.

$$WT_{start} := \sum_{i=1}^{x} playbackStartTime(p_i) - sceneChangeTime(p_i) \tag{7.6}$$

- P_{sum}: The number of pauses during playback of the whole video is defined as the count $pauses$ of times during individual scenes, where the frame f_{i+1} is loaded after frame f_i in more than $\frac{1000}{r} sec$ in Equation 7.7.

$$P_{sum} := \sum_{i=1}^{x} pauses(p_i) \tag{7.7}$$

- $A_{skipped}$: The number of frames in the whole video where annotations were not displayed because of the priority settings is expressed by this metric. It is the count $skippedFrames$ of times where an annotation a_o of frame f_i was not displayed during a scene (see Equation 7.8).

$$A_{skipped} := \sum_{i=1}^{x} skippedFrames(p_i) \tag{7.8}$$

- $DL_{not\ watched}$: The amount of data which was downloaded, but not displayed by the player is defined in Equation 7.9.

$$DL_{not\ watched} := \sum_{i=1}^{|\mathcal{DXN}_\mathcal{V}|} s(\pi_1(e_i, x_i)),\ (e_i, x_i) \in \mathcal{DXN}_\mathcal{V} \tag{7.9}$$

- $RDLV$: The data volume of elements that where downloaded more than once (because they where deleted from the cache, but are needed again) is defined in Equation 7.10.

$$RDLV := \sum_{i=1}^{|\mathcal{DXR}_\mathcal{V}|} s(\pi_1(e_i, x_i)),\ (e_i, x_i) \in \mathcal{DXR}_\mathcal{V} \tag{7.10}$$

- DLV: The quantity of downloaded data during the whole playback is defined in Equation 7.11.

$$DLV := \sum_{i=1}^{|\mathcal{DX}_\mathcal{V}|} s(\pi_1(e_i, x_i)),\ (e_i, x_i) \in \mathcal{DX}_\mathcal{V} \tag{7.11}$$

If an annotated interactive non-linear video is watched on a desktop-PC, low values for WF_{start}, WT_{start}, and P_{sum} are intended. This may imply higher values for $DL_{not\ watched}$ and $RDLV$, because with high bandwidth capacities and larger caches, future elements can be cached. If the video is watched on a smart phone, low values for $DL_{not\ watched}$ and $RDLV$ may be stipulated because of download limitations using a mobile phone contract with limited data plan for Internet usage. As a result, lesser future elements will be cached and the user will have to wait longer (higher values for WT_{start}) and maybe accept more breaks in the video flow (higher value for P_{sum}). (This section (7.1 Performance Metrics) was adapted from our previous work [MH12].)

7.2. Test Patterns and Test Graphs

We evaluate our algorithms and strategies in two ways. A more detailed evaluation with smaller video patterns is used to be able to make statements about the algorithms and strategies in standard or extreme cases. Larger scenarios which mix the patterns are used to verify the statements for real world scenarios. The remainder of this section illustrates and describes the patterns and scenarios for the evaluation.

7.2.1. Patterns in Annotated Interactive Non-linear Videos

Different patterns can be found in an annotated interactive non-linear video. Due to the high similarity of this kind of video with hypertext documents, we reviewed the work of Bernstein [Ber98] for suiting patterns. Some of the suggested patterns cannot be used for our evaluation, because they describe spatial relationships or relations which cannot be reproduced with a set of scenes. Each pattern consists of a certain structure (structure pattern) as well as a path which is taken by a viewer (user pattern). The structure pattern is the basis for the download strategies presented in Section 6.4. Both, structure and user pattern can be taken into account in the delete strategies introduced in Section 6.5. Figures 7.1 to 7.5 show the structure patterns as black framed rectangles linked by black arrows. The user patterns used in this work are indicated as blue filled rectangles linked by blue arrows. The user pattern is always the same for each structure pattern, but the assigned probabilities for choosing a path may vary from test case to test case in our evaluations. Suitable patterns are:

Sequence pattern: The sequence pattern is a linear pattern of scenes (see Figure 7.1). Each scene of the succession has to be loaded and played. Thus, sequence and user pattern are identical. Scenes or parts of the scenes may be skipped by jumping forward, but no navigation is provided due to its linear structure by this pattern. The sequence pattern is the only pattern used in our tests, that is not described by Bernstein [Ber98]. This pattern appears in scenarios where originally linear videos are cut into scenes which are then tagged with keywords and linked with a table of contents. The video can be watched as it was initially designed, but it is also possible to jump to certain scenes using the table of contents or the keyword search.

Figure 7.1.: Sequence pattern (structure and user pattern).

Cycle pattern: The cycle pattern [Ber98] is a linear succession of scenes which forms a loop (see Figure 7.2). Each scene of the cycle has to be loaded and played once, if the cycle is selected by the viewer. Navigation is provided after the scene where the cycle starts. The viewer can select to view the scenes of the cycle, for example to get more information about certain facts, or to proceed watching the main strand of the video without getting further information. Consequently, the user pattern may either have only two scenes (if the cycle is skipped), the same number of scenes as the structure pattern (if the cycle is viewed once), or $2 + n \cdot numberOfScenesInCycle$ scenes if the viewer enters the cycle more than once (n times). This is the only pattern where the user pattern may have more scenes than the structure pattern, if the user is not allowed to jump backwards and reselect a path. The performance of the delete strategies can be tested with this pattern for small cache sizes due to the possibility of returning to an already played scene which may then be played again. Therefor, the elements of the cycle scenes should have been kept in the cache. This pattern is useful in e-learning scenarios where certain learning contents can be explained in more detail. The user can decide if further explanations should be presented and chose to view the additional scenes in the cycle. The viewer can proceed watching the main strand of the video if she/he understood the learning contents.

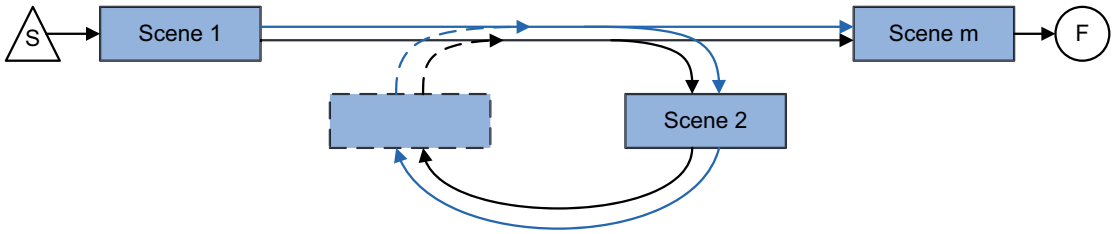


Figure 7.2.: Cycle pattern (structure and user pattern, based on [Ber98]).

Split/Join pattern: The split/join pattern [Ber98] is shown in Figure 7.3. One specific scene has to be watched after a set of scenes from which the user can select one. This structure assures that every second scene is watched by all viewers. Every other second scene can be selected individually. In the minimum case of this pattern, the viewer can select a scene once after the start scene and has to watch the final scene at the end. The user pattern always has less scenes than the structure pattern, if the user is not allowed to reselect, because only one of the scenes can be selected at a fork. These patterns may be used in scenarios where different strands of content are presented to the viewer, for example in sport events like skiing. Only one camera is available for start and finish of a run, but the viewers are able to select which camera perspective they want to see during a single run (for example helmet cam, drone cam, or a mixture of cameras along the route).

Mirrorworld/Counterpoint pattern: The mirrorworld pattern [Ber98] consists of two parallel strands of scenes (see Figure 7.4). A change between the strands can always happen at the end of a scene. Thereby, the viewer selects which strand of the video should be proceeded with. In contrast to the split/join pattern, each scene exists for the two views and the viewer can switch between the two strands after each scene. The user pattern always has $numberOfLayers + 2$ scenes compared to the structure pattern. This pattern can be used when a story is presented from two points of view, for example from the view of the persecutor and from the view of the victim in a film about a manhunt.

Sieve pattern: The sieve pattern [Ber98] describes a tree structure with branching paths (see Figure 7.4). A decision for the next scene is made after each scene. The number of scenes in the structure pattern is much higher than the number of scenes in the user pattern. Thus, the efficiency of download strategies is very important for this pattern, because a wrong strategy or a wrong parameter set may either lead to a unnecessarily high download volume or to an increased amount of waiting time at the beginning of a scene. This structure is part of many tour videos where the viewer is able to chose what should be shown next.

7.2.2. User Generated Scenarios

In addition to the test with the previously described patterns, we test our algorithms and strategies with user generated scenarios. Therefor, an experiment was conducted where four potential authors were asked to draw scene graphs according to a set of rules. The description of the tasks, which was given to the participants of the test, can be found in Appendix F, Figure F.1.

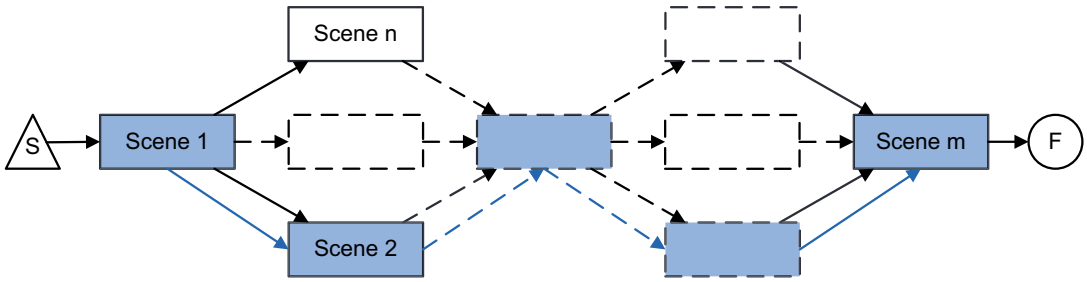


Figure 7.3.: Split/Join pattern (structure and user pattern, based on [Ber98]), hereafter "split pattern".

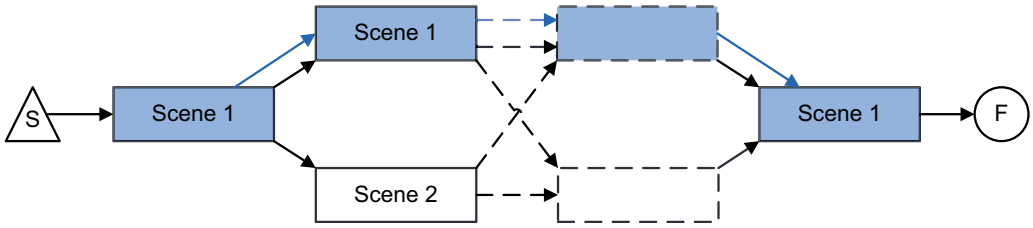


Figure 7.4.: Mirrorworld/Counterpoint pattern (structure and user pattern, based on [Ber98]), hereafter "mirrorworld pattern".

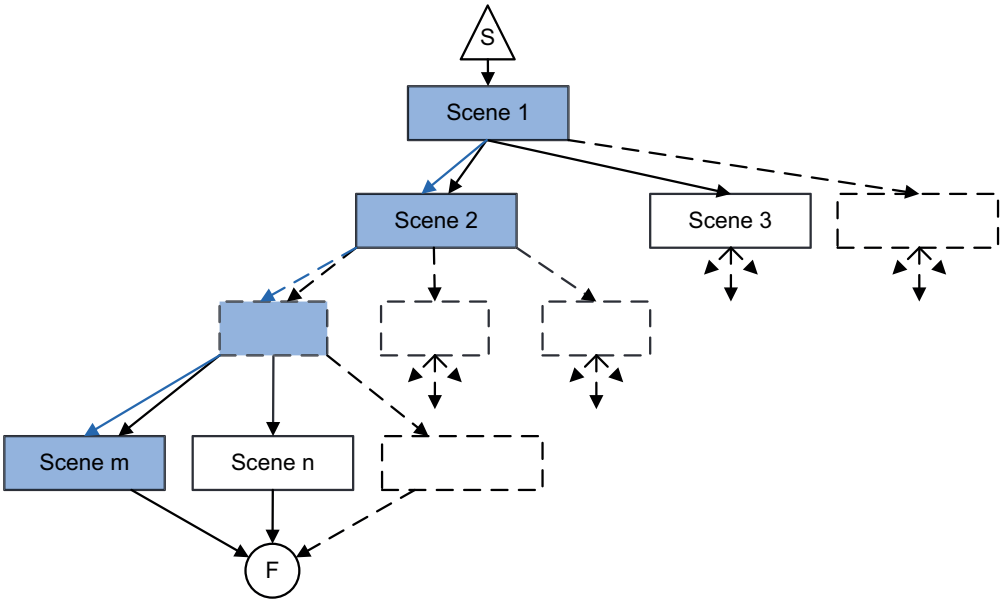


Figure 7.5.: Sieve pattern (structure and user pattern, based on [Ber98]).

1. The first task was the creation of a scene graph according to the following rules: The start element has no ingoing edges and one edge to the first scene. One scene may have one or more ingoing edges and has exactly one outgoing edge. A fork has one or more ingoing edges and two or more outgoing edges. The end of the video has one or more ingoing edges but no outgoing edge. One annotation can be linked to one or more scenes. These rules reflect the formal Definition 5.11 in Section 5.1 as well as the XML schema described in Section 3.1.
2. The second task was the selection of three not necessarily disjoint combinations of resolution, color depth, bandwidth, and cache size for the currently drawn scene graph. Annotation and scene sizes were calculated from the evaluation settings and scene durations afterwards. The drawn scene graph was given to the right neighbor after completing the task.
3. In the third task, the participants were asked to add probabilities to the outgoing edges of fork elements, which sum up to 100% for each fork. The scene graph was given to the right neighbor once more.
4. The forth and last task was to paint five different paths from start to end into the currently received scene graph.

Questions during the experiment were answered immediately and aloud, so all participants had the same information during the whole time of the experiment.

7.3. Test Configurations

One test configuration is composed of three kinds of settings for download and cache management for annotated interactive non-linear video. These are detailed in the following and will be described and formalized in the remainder of this section in more detail:

- ***Description of the video***: It includes the definitions of scenes and corresponding frames, of annotations and where they are shown and hidden, as well as the structure of the scene graph. The probabilities of choosing a scene at a fork have to be determined and are fixed for the whole simulation to provide comparable and repeatable results. We set the values for our scenarios using realistic or more extreme values based on potential real world values. Furthermore, the way taken through the annotated interactive non-linear video by the current user has to be set.
- ***Parameters for the strategies***: The variables for the video playback, download scheduling and delete strategies have to be configured in order to customize and select a set of strategies.
- ***Description of the environment***: Cache size and bandwidth have to be defined and all combinations of them need to be tested.

7.3.1. Description of the Videos and the User Behavior

The description of a video consists of the definition of the duration of single scenes, of the number of annotations in a scene, and of the resolution of video and annotations. Further-

more, relationships between the scenes which form a scene graph need to be defined as well as probabilities for the selection of certain scenes at a fork.

7.3.1.1. Descriptions of the Patterns

Besides the already mentioned criteria, another criterion for the test cases with the patterns is the overall playback duration. We chose to test our patterns with a playback duration of the user pattern of about five minutes, based on a review of average video lengths in online videos. These range from "5.1 minutes in February 2011" [Han11] to "4 minutes and 12 seconds" for a YouTube video in 2009 (July to December) [Sys10]. The average length of the most popular YouTube videos from January 2011 to March 2012 was 3 minutes and 53 seconds [Pro12]. The duration of the playback may vary slightly due to the design of the structure patterns.

The description of a single test consists of five different variables that have to be set. The variables are a pattern, the probabilities for selecting a scene, the duration of a scene, the number of annotations, and the size of the media which correlates with its resolution. First, a **pattern** with a defined width v (or cycle length of v scenes) has to be selected from the set of patterns as defined in Set 7.12. :

$$\mathcal{P}atterns := \{Cycle_v,\ Mirrorworld_v,\ Sieve_v,\ Split_v,\ Sequence\},\ v > 1, v \in \mathbb{N} \quad (7.12)$$

After selecting a pattern which only defines the structure of the test case, the precise appearance of each scene as well as the **probability for selecting a path/button** has to be determined. The set of probability distributions is defined in Set 7.13. Thereby, $prob_{avg}(p_x, w_{x,i})$ denotes the setting where each path/button at a selection is chosen with the same probability as defined in Equation 7.16. The $prob_{worst}(p_x, w_{x,i})$ and $prob_{worstavg}(p_x, w_{x,i})$ settings assign equal and comparable high percentages to all paths except the path w_{x,i_1}. We calculate the distribution of the percentages for $prob_{worstavg}(p_x, w_{x,i})$ as defined in Equation 7.17. For $prob_{worst}(p_x, w_{x,i})$, the denominator is squared as defined in Equation 7.18. The remaining percent are distributed equally to all other scenes. The $prob_{best}(p_x, w_{x,i})$ and $prob_{bestavg}(p_x, w_{x,i})$ settings assign equal and comparable low percentages to all paths except the path w_{x,i_1}. Thereby, all paths except the path w_{x,i_1} get the same percentages as the path with the lowest priority in the $prob_{worst}(p_x, w_{x,i})$ setting in $prob_{best}(p_x, w_{x,i})$. The remaining percent are assigned to the path/button w_{x,i_1} with the high probability (see Equation 7.14). The $prob_{bestavg}(p_x, w_{x,i})$ setting halves the denominator (compared to $prob_{bestavg}(p_x, w_{x,i})$) of each fraction as defined in Equation 7.15.

$$\begin{aligned}\mathcal{P}robabilities := \{&prob_{best}(p_x, w_{x,i}),\ prob_{bestavg}(p_x, w_{x,i}),\ prob_{avg}(p_x, w_{x,i}),\\ &prob_{worstavg}(p_x, w_{x,i}),\ prob_{worst}(p_x, w_{x,i})\}\end{aligned} \quad (7.13)$$

$$prob_{best}(p_x, w_{x,i}) := \begin{cases} 1 - \frac{|\mathcal{P}_{succ(p_x)}|-1}{|\mathcal{P}_{succ(p_x)}|^2}, & \text{path/button } w_{x,i_1} \text{ with highest probability} \\ \frac{1}{|\mathcal{P}_{succ(p_x)}|^2}, & \text{other paths} \end{cases} \quad (7.14)$$

$$prob_{bestavg}(p_x, w_{x,i}) := \begin{cases} 1 - \frac{|\mathcal{P}_{succ(p_x)}|-1}{\frac{1}{2}|\mathcal{P}_{succ(p_x)}|^2}, & \text{path/button } w_{x,i_1} \text{ with high probability} \\ \frac{1}{\frac{1}{2}|\mathcal{P}_{succ(p_x)}|^2}, & \text{other paths} \end{cases} \tag{7.15}$$

$$prob_{avg}(p_x, w_{x,i}) := \frac{1}{|\mathcal{P}_{succ(p_x)}|} \tag{7.16}$$

$$prob_{worstavg}(p_x, w_{x,i}) := \begin{cases} \frac{1}{\frac{3}{2}|\mathcal{P}_{succ(p_x)}|}, & \text{path/button } w_{x,i_1} \text{ with low probability} \\ \frac{(1-\frac{1}{\frac{3}{2}|\mathcal{P}_{succ(p_x)}|})}{(|\mathcal{P}_{succ(p_x)}|-1)}, & \text{other paths} \end{cases} \tag{7.17}$$

$$prob_{worst}(p_x, w_{x,i}) := \begin{cases} \frac{1}{|\mathcal{P}_{succ(p_x)}|^2}, & \text{path/button } w_{x,i_1} \text{ with lowest probability} \\ \frac{(1-\frac{1}{|\mathcal{P}_{succ(p_x)}|^2})}{(|\mathcal{P}_{succ(p_x)}|-1)}, & \text{other paths} \end{cases} \tag{7.18}$$

Figure 7.6 shows the different distributions from Equations 7.14 to 7.18 per scene at one fork. The light blue bar indicates the part of the whole download volume which is assigned to the scene p_{i_1} which will be selected by the viewer for the chosen distribution. Depending on the number of scenes at a fork, the bars in the other colors indicate the download volume of each of the scenes.

We furthermore distinguish between the **scene durations** dur_{short}, dur_{medium}, and dur_{long} (set by the length of the scene ($dim(p_x)$, see Function 5.2) divided by the frame rate r (see Function 5.1). A combination of the durations, dur_{combi}, is used as well and expressed by a regular expression. The combination of durations dur_{combi} assigns the three defined durations to the scenes in ascending order with repetitions until the pattern length is reached. The set of scene durations is defined in Set 7.19. The durations of the scenes are calculated in relation to the overall playback duration dur_{pb} of the test scenario. They are presented in Equation 7.20. Using a playback duration of five minutes, the length of a short scene is 15 seconds, a medium scene is thirty seconds, and a long scene is one minute. These values are derived from real world videos and adapted that they are a factor of dur_{pb} to be able to create user pattern of five minutes.

$$SceneDuration := \left\{dur_{short},\ dur_{medium},\ dur_{long},\ dur_{combi}\right\} \tag{7.19}$$

$$\begin{aligned} dur_{short} &:= \frac{dur_{pb}}{20} \text{ sec} \\ dur_{medium} &:= \frac{dur_{pb}}{10} \text{ sec} \\ dur_{long} &:= \frac{dur_{pb}}{5} \text{ sec} \\ dur_{combi} &:= \left[dur_{short}, dur_{medium}, dur_{long}\right]^+ \end{aligned} \tag{7.20}$$

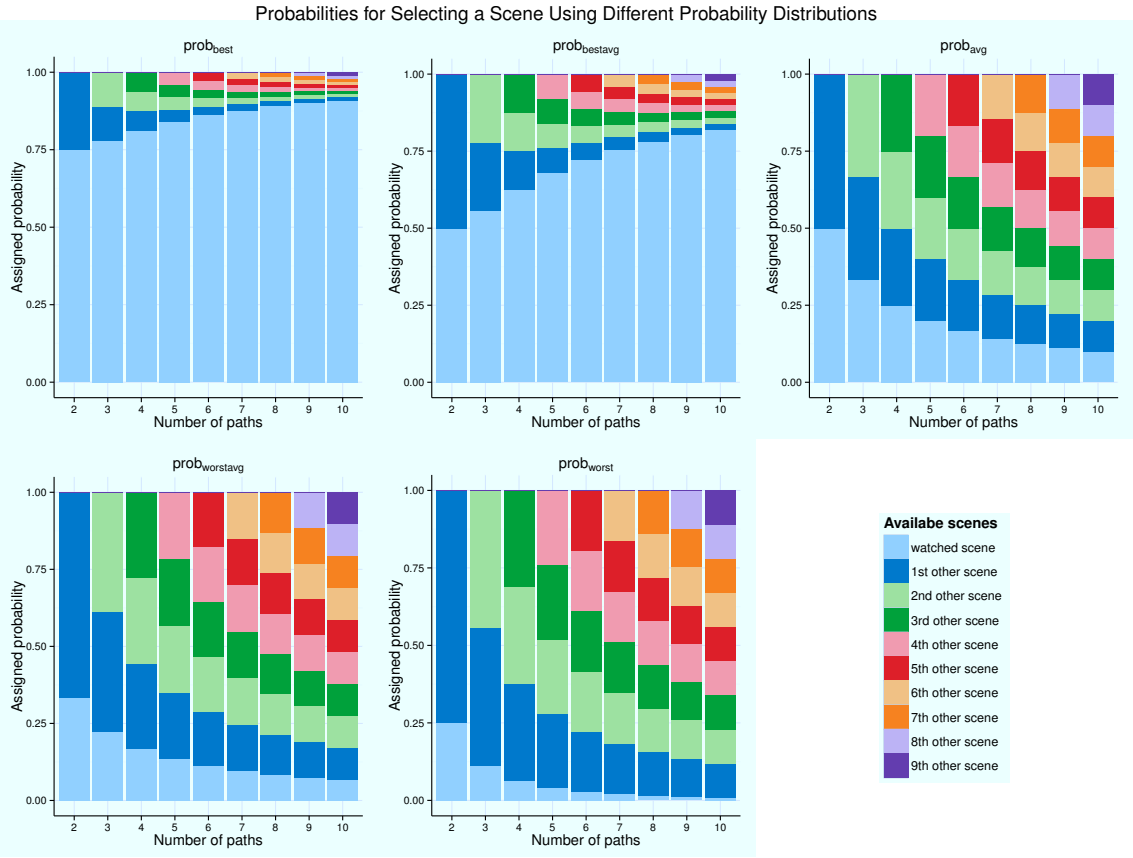


Figure 7.6.: Probabilities for selecting a scene using different probability distributions.

We assume that the annotations are distributed equally over the duration of a scene and define the amount of annotations for all scenes as ac_{few}, $ac_{fewmedium}$, ac_{medium}, $ac_{mediummany}$, and ac_{many}. These settings form the **number of annotations** $\mathcal{A}nnoCount$ as in Set 7.21. The number of annotations per scene is determined by the length of a scene and the time lag between two annotations (see Equations 7.22). The first annotations is always shown with the first frame of a scene. Displayed annotations are hidden when a new annotation is triggered.

$$\mathcal{A}nnoCount := \left\{ac_{few},\ ac_{fewmedium},\ ac_{medium},\ ac_{mediummany},\ ac_{many}\right\} \tag{7.21}$$

$$\begin{array}{lcl} ac_{few} & := & 1 \text{ annotation per scene} \\ ac_{fewmedium} & := & 1 \text{ annotation every 250 frames} \\ ac_{medium} & := & 1 \text{ annotation every 100 frames} \\ ac_{mediummany} & := & 1 \text{ annotation every 25 frames} \\ ac_{many} & := & 1 \text{ annotation every 10 frames} \end{array} \tag{7.22}$$

The last set of definitions for a scenario are the sizes (as defined in Function 5.5) of the used media as defined in Set 7.23. The **size** settings use rounded values derived from real world screen sizes $1920x1080$ for res_{high}, $1280x720$ for res_{medium}, and $720x480$ for res_{low} using a color depth of $8bits/color$ and an RGB $4:4:4$ color model. This results in the values given in Equation 7.24. We use no compression for our calculations, so each frame of a video has the same size.

$$Sizes := \left\{size_{small},\ size_{medium},\ size_{large}\right\} \tag{7.23}$$

$$\begin{aligned} size_{small} &:= \begin{cases} 1 \text{ MB, per frame} \\ 10 \text{ MB, per annotation} \end{cases} \\ size_{medium} &:= \begin{cases} 2,5 \text{ MB, per frame} \\ 25 \text{ MB, per annotation} \end{cases} \\ size_{large} &:= \begin{cases} 6 \text{ MB, per frame} \\ 60 \text{ MB, per annotation} \end{cases} \end{aligned} \tag{7.24}$$

7.3.1.2. Scenarios

The scenarios used in this work are the result of the experiment described in Section 7.2.2. The produced scene graphs from the first task can be found Figure 7.7 and in Appendix F, Figures F.2 on page 257, F.3 on page 259, and F.5 on page 262. The outcome of the second task can be found under "Environment settings" in Table 7.1 and in Appendix F, Tables F.1 on page 258, F.2 on page 261, and F.3 on page 263. As a third task, the participants added probabilities to the outgoing edges of fork elements. The results thereof are shown in Figure 7.7 and in Appendix F, Figures F.2 on page 257, F.3 on page 259, and F.5 on page 262. The selected paths of the last task are illustrated in Figure 7.7 and in Appendix F, Figures F.2 on page 257, F.4 on page 260, and F.5 on page 262.

7.3.1.3. User Behavior

We assume the following user behavior for all of our tests: The user interacts with the video at forks in the video flow. The viewer does not jump from scene to scene, for example by using the table of contents or the keyword search. No forward or backward jumps are made in a scene. Furthermore, neither going back in the history of reviewed scenes nor reselections at forks are possible. Accordingly, only forward scene selections are evaluated hereafter. We chose this behavior, because it requires a certain amount of interaction but allows us to test our algorithms and strategies in a controlled environment at the same time.

The evaluation of some other actions from Section 5.3 is a task for future work. These include slow and fast-forward, jump forward and backward, as well as navigation with the table of contents and the keyword search. With a large enough cache size, play backward, slow and fast rewind in a scene, as well as pause are covered by our algorithms and strategies and would lead to improved results due to a longer available download time for future elements in most of the cases. Additional algorithms and strategies would be necessary for this behavior when a small cache size is used. Pan, tilt, and zoom are strongly influenced by their implementation and thus their requirements for algorithms and strategies for download and cache management. They may also be studied in more detail in future work.

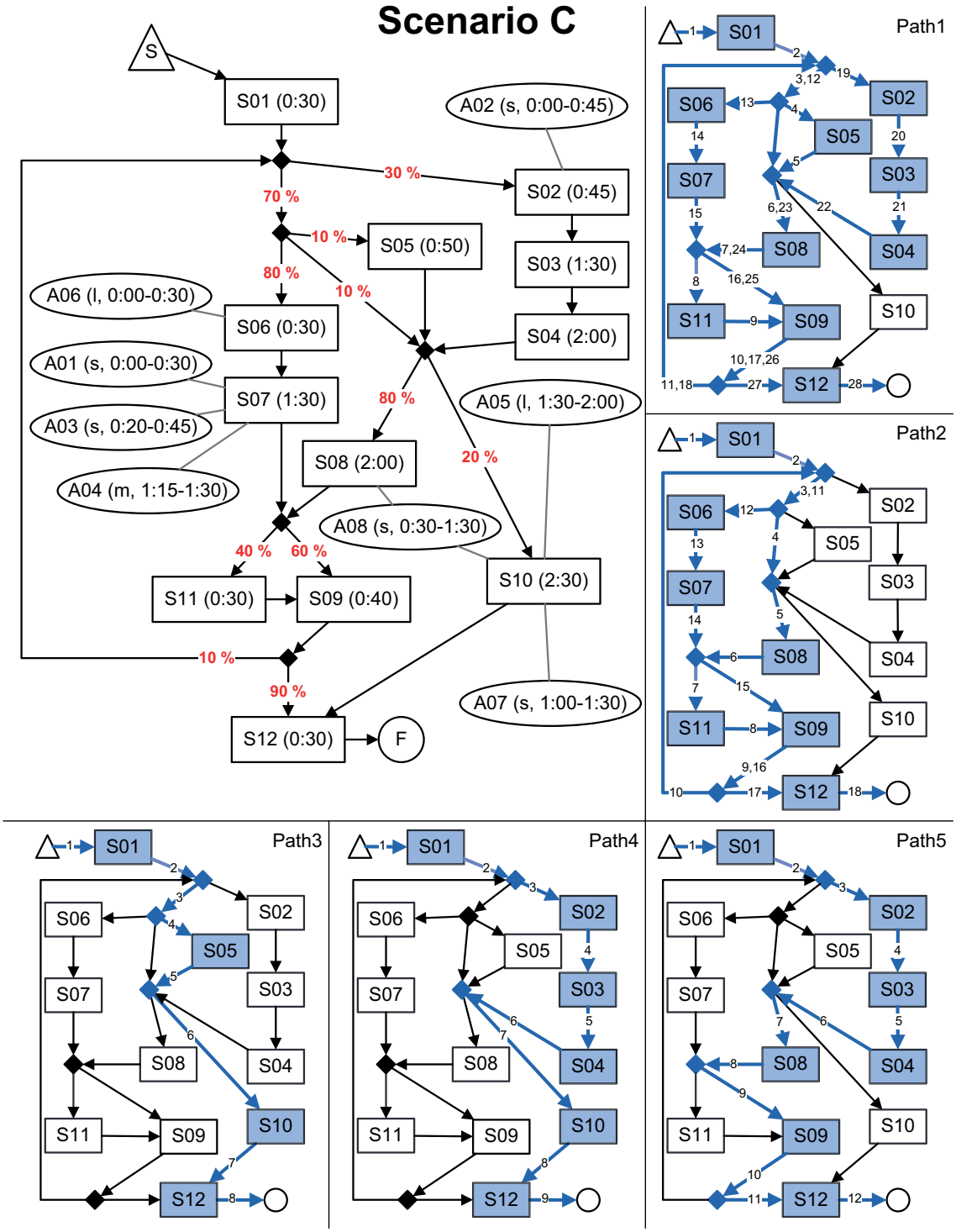


Figure 7.7.: Scenario C: scene graph with probabilities and annotations (top left) and five paths through the graph.

Settings for scenario C				
Environment settings				
Environment name	Resolution	Color depth (bit)	Bandwidth (Mbit/s)	Cache (MB)
E1	2560 x 1440	24	25	1024
E2	1680 x 1050	16	16	2048
E3	1366 x 768	32	16	64
Annotation sizes (KB)				
Annotation name	Size	E1	E2	E3
A01, A02, A03, A07, A08	small	21600	6890,625	8196
A04	medium	108000	34453,125	40980
A05, A06	large	270000	86132,8125	102450
Scene sizes (MB)				
Scene name	Duration (sec)	E1	E2	E3
S01, S06, S11, S12	30	7910,16	2523,42	3001,46
S02	45	11865,23	3785,13	4502,20
S03	90	23730,47	7570,27	9004,39
S04	120	31640,63	10093,69	12005,86
S05	50	13183,59	4205,70	5002,44
S07	90	23730,47	7570,27	9004,39
S08	120	31640,63	10093,69	12005,86
S09	40	10546,88	3364,56	4001,95
S10	150	39550,78	12617,11	15007,32
Path sizes and durations				
Path name	Duration (min:sec)	E1	E2	E3
Path1	14:35	177978,52	56777,00	67532,96
Path2	6:50	97558,59	31122,21	37018,07
Path3	4:20	68554,69	21869,66	26012,70
Path4	7:45	122607,42	39113,04	46522,71
Path5	7:55	125244,14	39954,19	47523,19

Table 7.1.: Scenario C: settings for the environment and calculations for the annotation sizes, the scene sizes, and the path sizes and durations.

7.3.2. Parameters for the Algorithms/Strategies

The second category of settings are the parameters for the strategies. One strategy with according parameter settings has to be chosen from each of the sets $PlaybackStart$, $Prefetch$, and $Delete$ (see Definitions 7.25, 7.26, and 7.27) for a single test. The definitions and descriptions of the strategies can be found in Section 6. Each configurable strategy is tested with different parameters as defined with each set of strategies.

$$\begin{aligned} \mathcal{P}laybackStart := &\{PLAY\_SCENE,\ PLAY\_MIN\_REL(f_m),\\ &PLAY\_MIN\_REL\_PRIO(f_{m/n}, \Lambda),\ PLAY\_STARTUP(f_x)\},\\ &\Lambda \in \{1,2\},\\ &f_x \in \{125, 250, 375\} \end{aligned} \tag{7.25}$$

$$\begin{aligned} \mathcal{P}refetch := &\{PREFETCH\_SL(z_{SL}, \gamma, \Lambda, dist)\},\\ &z_{SL} = 1,\\ &\gamma \in \{|p_x|, m, n\},\\ &\Lambda \in \{1,2\},\\ &dist \in \{0,1,2,3\} \end{aligned} \tag{7.26}$$

$$\begin{aligned} \mathcal{D}elete := &\{DELETE\_SD(\mu),\ DELETE\_LRU,\ DELETE\_D\_PROB,\\ &DELETE\_PRIO(\Lambda)\},\\ &\mu \in \{0, 0.5, 1\},\\ &\Lambda \in \{1,2\} \end{aligned} \tag{7.27}$$

7.3.3. Description of the Environment

We define two sets for environmental settings in our test scenarios, the set of cache sizes in Set 7.28 and the set of bandwidths in Set 7.29. The cache sizes are inspired by commonly used memory sizes. The used bandwidths are based on currently available network bandwidths: 5,76 Mbit/s (HSPA), 10 Mbit/s (10-Mbit/s-Ethernet), 16 Mbit/s (ADSL2+), 25 Mbit/s (VDSL25), 32 Mbit/s (Cable-Internet 32), 50 Mbit/s (VDSL50), 100 Mbit/s (10-Mbit/s-Ethernet)[1].

$$\mathcal{C}acheSize := \{512\,MB,\ 1024\,MB,\ 4096\,MB,\ 16384\,MB,\ 32768\,MB\} \tag{7.28}$$

$$\begin{aligned} \mathcal{B}andwidth := &\{5,76\,Mbit/s,\ 10\,Mbit/s,\ 16\,Mbit/s,\\ &25\,Mbit/s,\ 32\,Mbit/s,\ 50\,Mbit/s,\ 100\,Mbit/s\} \end{aligned} \tag{7.29}$$

[1] `http://www.heise.de/netze/tools/bandbreitenrechner/` (accessed July 22, 2013)

7.4. Simulation Environment

In order to investigate several strategies for a download and cache management for annotated interactive non-linear video players, a simulation environment was implemented to be independent from current platform and browser implementations. The focus of our simulation is on the internal processes in the player, not on network transmission. For the latter are different tools available. Network simulation tools like NS-2 [The], OPNET [Riv13], and OMNet++ [OMN13] are used for simulation of protocols and network behavior. Several extensions and simulation results in the area of video referencing these tools exist. They are mainly dealing with the transmission of linear videos over a network. Other work, not implemented in one of these tools, can be found on internal procedures in architectures for multi-core video decoding, which are simulated by Seitner et al. in [Sei+09]. Klaue, Rathke, and Wolisz present EvalVid, "a complete framework and tool-set for evaluation of the quality of video transmitted over a real or simulated communication network" [KRW03]. An integration of Evalvid and NS-2 is presented by Ke et al. in [Ke+06]. A simulation for rate adaption in streaming of multimedia content, called Evalvid-RA, is outlined by Lie and Klaue in [LK08]. It is also used in combination with the NS-2 tool. Boronat, Montagud, and Vidal describe a "full RTP/RTCP implementation for NS-2" as well as the combination with the EvalVid Framework [KRW03] or VLC[2] [BMV10]. A simulation study which streams video over mobile ad hoc networks is presented by Chow and Ishii in [CI06]. These tools deal with streaming and decoding of videos which is not our focus. Consequently, the tools would have needed a full extension with our algorithms and strategies while dealing with the specialized streaming or decoding algorithms. Gaggi and Celentano describe an authoring environment for complex multimedia presentations which contains an execution simulator to check the behavior of the presentation [GC02]. Neither download nor cache management is part of this work, which focuses only on "the synchronization [of] relationships among media" [GC02].

With the outlined shortcommings of the described frameworks, we decided to implement the simulation environment from scratch, because none of the available simulation tools provides a suitable framework for the objectives of our evaluation. Another reason for the implementation from scratch is that the authoring tool illustrated in Section 3.2, is written as an ERCP-application in Java. In order to integrate the software introduced in this work via the plug-in concept into the ERCP-application in future work, we needed to write our simulation environment in Java and could not use one of the existing simulation toolkits mainly implemented in C or C++.

To provide a flexible environment for our simulation, we implemented a modular simulation framework for annotated interactive non-linear video. It provides interfaces for exchangeable algorithms and strategies for download scheduling and cache management. XML files for the configuration of the simulation, the user behavior and the structure of the video can be created manually or by a generator application. They are then processed by the framework to control each simulation. As already described in Section 6.1, the simulation framework consists of two main components, a simple server and a client with the download and buffer logic (see Figure 6.1 in Section 6.1). This model is now described from a more technical point of view and how it is implemented in the simulation framework. We furthermore give an overview on how the simulation settings are saved for the single test runs and which command line functions are provided.

[2] `http://www.videolan.org/vlc/index.html` (accessed July 25, 2013)

7.4.1. Internal Structure of the Framework

The framework consists of a player and a server as outlined in Section 6.1. The **server** receives a request object from the client and responds with the relevant data. Currently we use a virtual server which does not use real networking but simulates a fixed data rate and allows local file access. The **client** consists of a download part, a delete part, a buffer control, a scenario handler, and a player part. There is a ***download agent*** with customizable download threads (e.g. using one thread means serial download) for downloading certain resources. Threads are configurable for downloading only certain types of resources (e.g. one thread for images) or for downloading queues provided by the download scheduler. To determine which resources have to be downloaded next, we use a bunch of interchangeable download strategies implemented in the scene scheduler. The resources are also affected by users choices at forks in the video flow. The current implementation two ***buffers***/***caches***, a download buffer and a player cache. Downloaded resources are stored in the download buffer first and then put into the player cache. Delete schedulers decide which resources have to be deleted from the cache by the ***cleaning agent*** if a certain level is reached. The ***player*** uses actions for logical control, e.g. a show annotation action, a control action, or a load video action. New actions can be added by implementing the required interfaces. Furthermore, decoder and player are exchangeable. Currently only a fixed decoding time is used. A ***scenario handler*** reads the whole XML file at the beginning of the simulation and knows the structure of the whole video after that. As the algorithms need to know the progression of videos/sequences of scenes, a scene graph is built up. It is used to find predecessor and successor of a certain scene as well as all annotations attached to a scene. All other components obtain the necessary information from the scenario handler. Actions are mainly separated into triggered actions (actions executed at a certain frame), user actions (simulated user interaction, e.g. play/pause or selection of the next scene), and system actions (e.g. load next scene).

7.4.2. Simulation Settings

Our simulation settings are stored in five separate XML files for one test run plus one control file which combines these five files for an automated simulation execution of a set of test runs. These files can be described as follows:

- ***Settings file:*** The settings file describes commonly needed variables for the simulation. It defines all parameter settings as well as the settings for the simulation tool according the logging behavior and data collection. Furthermore, it defines the speed-up (compared to real time) for the simulation.
- ***Environment file:*** Combinations of bandwidth and cache size are stored in the environment file.
- ***Scenario file:*** A scenario file describes the interactive video itself. The basic definitions of this file are a list of scenes defining the structures from Section 7.2.1 and the storyboard for the scenes, which defines what annotations should be shown and when. It furthermore defines the lengths of the scenes and the annotation and frame sizes.
- ***PathSet file:*** This file contains the definition of the user behavior. In order to rerun the simulation with the same path through the video and equal click times of the user,

the choice at each of the forks is defined with an adjustable click time. Furthermore, probabilities for taking a certain path are defined in this file.

- ***Strategy file:*** This file specifies which combination of strategies is used for a simulation. The strategies are chosen and their variables are set to the required values.

The sixth file is a ***control file*** which lists combinations of the other files for an automated execution of the single test runs.

7.4.3. Implementation of the Framework

The simulation framework is implemented in Java [Ora]. It consists of two parts, an XML generator and a simulation tool. The XML generator creates the six types of XML files described in Section 7.4.2 for the pattern-based test cases. It is also possible to create the settings file, the strategy file, and the control file for the scenario-based tests. Thereby, the manually created environment files, scenario files and pathSet files are linked together. The created files are the input for the simulation tool, which evaluates the therein described test cases and generates result files. Both tools are described in detail hereafter. Parts of the implementation used in the simulation tool (7.4 Simulation Environment) are based on the Masters Thesis of Jürgen Hoffmann [Hof12].

7.4.3.1. XML Generator

The XML generator creates a file for each defined parameter value combination which is meaningful for this type of file. This process results in one settings file and different files for the environment, the scenarios, the paths, and the strategies. The names of all created files are saved in a separate list for each file type. After the generators for the single files have finished, a last generator iterates over the lists and adds the different resulting combinations of files to one or more control files. If a scenario test is generated, manually created files are added to a list and are then linked with the automatically generated settings and strategy files.

7.4.3.2. Simulation Tool

The simulation tool is used to perform the simulations defined in the XML files described in Section 7.4.2. Firstly, it reads the XML files and saves the values into internal data structures. Then it creates all necessary objects for the simulation from these data structures. After one run of a simulation is started, it collects result values needed for the metrics defined in Section 7.1. These are written to a CSV file when one simulation run is finished.

This tool is implemented for executing the different simulations parallel on one processor and with a speed-up compared to real time. Therefore, the tool was analyzed and tested for possible deadlocks, memory leaks, and other performance issues with Java profilers like

JProfiler[3], YourKit Java Profiler[4], and the built in tools from IntelliJ IDEA[5]. The simulation tool can be started in three different modes which all take different input parameters:

- **getpoolsize**
  This mode is used to determine the number of parallel threads that can be used without causing the results to be inaccurate because of calculations with larger run times. Therefor, an initial number of threads is set. A customized test file with significant test cases is used for this test. This file is simulated with the given number of threads, then the results are compared with each other. If all results for one metric do not extend beyond a given value, the selected number of threads can be used for the tested system. If the deviation of the results is to excessive, an algorithm similar to a binary search is used to find the largest possible number of threads that can be used. The command line for this mode is shown in Listing 7.1. The first argument (`getpoolsize`) is the selected action, the second argument (`initialPoolSize`) is the initial number of threads, the third argument (`nodeNumber`) is the node number of the cluster node, and the fourth argument (`runs`) defines how often the test should be repeated. The node number is important when the command is executed on different cluster nodes to see possible discrepancies between single nodes.

```
1 getpoolsize initialPoolSize nodeNumber runs
```

Listing 7.1: Command line arguments of the getpoolsize mode.

- **simulate**
  The `simulate` mode is used for simulating all created input files one after the other. The command line for this mode is shown in Listing 7.2. Input parameters are the pool size (`poolSize`) determined with `getpoolsize` for the used simulation hardware, the number of the cluster node (`nodeNumber`) the simulation is executed on, and the overall number of simulation nodes (`numberOfNodes`) used for the simulation. Each node simulates a subset of the control files which is defined by an initial number (`nodeNumber`) and an offset defined by the `numberOfNodes`.

```
1 simulate poolSize nodeNumber numberOfNodes
```

Listing 7.2: Command line arguments of the simulate mode.

- **resume**
  The resume mode can be used in case a simulation was aborted for some reason on a single cluster node. The result files in a given folder are analyzed and compared against the simulation control files. The simulation is then resumed from the first missing simulation run on. The command line for this mode is shown in Listing 7.3. The first input parameter is the selected action (`resume`), the second parameter (`folderName`) is the result folder for which the simulations should be resumed, the other parameters (`poolSize`, `nodeNumber`, `numberOfNodes`) are the same as described for the `simulate` action.

```
1 resume folderName poolSize nodeNumber numberOfNodes
```

Listing 7.3: Command line arguments of the resume mode.

[3] `http://www.ej-technologies.com/products/jprofiler/overview.html` (accessed April 26, 2014)
[4] `http://www.yourkit.com/overview/index.jsp` (accessed April 26, 2014)
[5] `http://www.jetbrains.com/idea/` (accessed April 26, 2014)

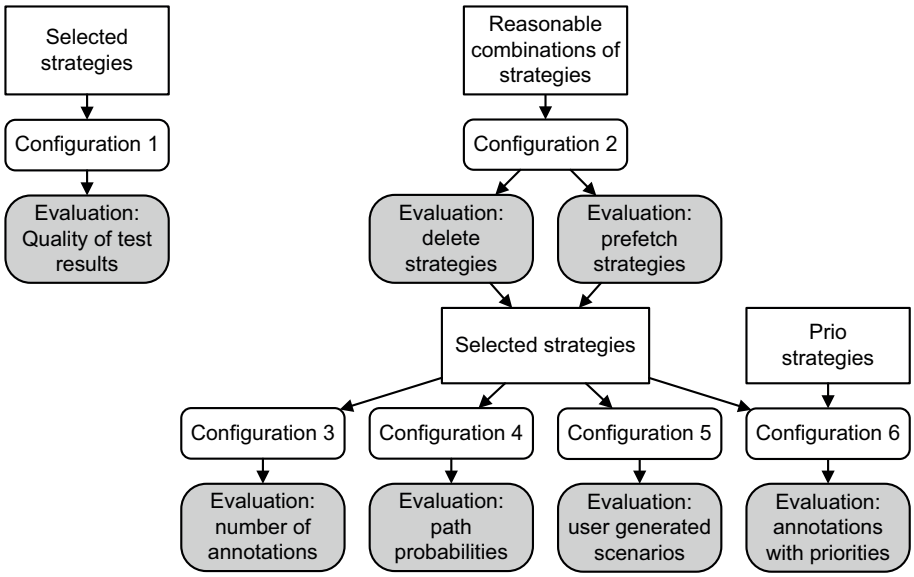


Figure 7.8.: The evaluation is performed in a two-part process: a small subset of the test cases/simulation settings is used to show the quality of the test results, then the evaluations are performed with larger combinations of simulation settings and strategy configurations.

7.4.3.3. Simulation Hardware

The simulations are carried out on a twelve node high performance computing cluster. Each node has two Intel Xeon QuadCore processors with 2 GHz clock rate, 16 GB RAM and two 74 GB SATA hard disks with a RAID 0 configuration. The master node is used to monitor the computing nodes during the simulations, it is not used for the calculations. We used `getpoolsize` to find a suitable combination of speed-up and pool size. As a result, the simulations in this work are run with a speed-up of twenty and a pool size of eight per node. This results in an about 2240 times faster execution of the simulation compared to the initial implementation described in [Hof12].

7.5. Performance Evaluation

We perform our evaluation in two steps. Firstly, we show that our test results are statistically significant, secondly we perform tests with varying environment settings to show the behavior of the strategies in different specialized situations. Figure 7.8 shows our evaluation plan, especially for the second and larger part of the evaluation. Thereby, we first use a wide range of strategies in an environment with an average setting for annotations, pattern width, media sizes, and scene lengths. As a result, suitable strategies can be selected for the further tests. These selected strategies are then tested in more specialized environment settings. We vary the number of annotations, the width of the $sieve_3$ pattern and assigned path probabilities, and use priorities for the annotations. Furthermore, we test the selected strategies with the user generated scenarios from the experiment described in Section 7.2.2. Unless otherwise stated, we always pre-fetch whole scenes for PLAY_SCENE and the part for playable for PLAY_MIN_REL and PLAY_STARTUP in the remainder of this work.

7.5.1. Quality of the Test Results

To show the distribution of the evaluation results, we start the simulation with a small but significant combination of strategies and environment configurations and repeat the run 1000 times. We want to show that the standard deviations and standard errors are small enough and the results between the strategies for one metric are distinct enough or the same to be able to make statements regarding the quality of the strategies with one single test run. This approach is important because due to the high number of test runs and the execution of the simulation in twenty times real time, one test set takes about three to four days to finish, even if eight times twelve tests are started in parallel (eight tests per node, twelve cluster nodes). This makes it impossible to run the simulation as often as needed to get statistically significant results. For this significance test, we use the configuration described in Equations 7.30 and 7.31.

$$pattern \times probability \times duration \times anno \times size \times cache \times bw \times pbstart \times pref \times del \tag{7.30}$$

$$\begin{aligned}
pattern &\in \{Cycle,\ Sieve_3,\ Sequence\} \subseteq \mathcal{P}atterns \\
probability &= prob_{avg}(p_x) \in \mathcal{P}robabilities \\
duration &= dur_{medium} \in \mathcal{S}ceneDuration \\
anno &= ac_{medium} \in \mathcal{A}nnoCount \\
size &= size_{medium} \in \mathcal{S}izes \\
cache &\in \{512MB,\ 4096MB,\ 16384MB\} \subseteq \mathcal{C}acheSize \\
bw &\in \{5,76Mbit/s,\ 25Mbit/s,\ 50Mbit/s,\ 100Mbit/s\} \subseteq \mathcal{B}andwidth \\
pbstart &\in \{PLAY_SCENE,\ PLAY_MIN_REL(f_m), \\
&\qquad PLAY_STARTUP(f_x)\} \subseteq \mathcal{P}laybackStart \\
&\qquad f_x = 250 \\
pref &\in \{PREFETCH_SL(z_{SL}, y, \Lambda, dist)\} \subseteq \mathcal{P}refetch \\
&\qquad z_{SL} = 1,\ y \in \{|p_x|, m\},\ \Lambda = 1,\ dist = 1 \\
del &= DELETE_SD(\mu) \in \mathcal{D}elete \\
&\qquad \mu = 1
\end{aligned} \tag{7.31}$$

The results for the metrics WF_{start}, WT_{start}, P_{sum}, $DL_{not\ watched}$, $RDLV$, and DLV of the 1000 test runs are then analyzed grouped by bandwidth and cache size for each of the tested patterns. We calculated the minimum, the maximum, the mean value, the median, and the standard deviation [FMF12, p. 39] for each test case. The results can be found in Appendix G, Tables G.1 to G.6. It can be summarized that the standard deviation is comparably small for the time/frame based metrics and may be high for the metrics concerning the download volume (especially for the $cycle_3$ and the $sieve_3$ pattern where points in time where elements are deleted from the cache is crucial).

Furthermore, we conducted a Kruskal-Wallis test (for descriptions of the test see [Kru52; KW52] and [SC56, pp. 206-216]) to show if the results of different groups are significantly different. Thereby the H-value was calculated as described in [Kru52, p. 526].

After that a multiple comparison test as post hoc test [She07, pp. 986-990] for all pattern/-cache size/bandwidth combinations where the result of the tested metric was not exactly equal or had standard deviations larger than zero, was conducted. Thereby, a p-Value of 0.0001 was used. We chose to use this non-parametric test, because the data did not meet the first two requirements for parametric tests, normality and homogeneity of variance [FMF12, p. 168]. Furthermore, the Wilcoxon's rank-sum test [She07, pp. 791-792] was not suitable because we dealt with more than two independent variables. The results of Kruskal-Wallis and the post hoc test can be found in Appendix H, Tables H.1 to H.6. Important findings of these tests will be discussed combined with the other statistical data hereafter. The histograms show the count of how often a test had a specific value as a result. The higher and the more narrow a bar is, the more uniform are the results. In the further course of this subsection (7.5.1 Statistics), the following abbreviations are used in the text indicating the described combinations of strategies:

- PS: PlayScene__PrefetchSL_whole_1
- PML: PlayMinReload__PrefetchSL_playable_1
- PSU: PlayStartup_10__PrefetchSL_playable_1

7.5.1.1. WF_{start} metric

Figure 7.9 shows the data distribution of the number of frames to wait WF_{start} at the beginning of scenes. The histogram for the sequence pattern shows about equal values for the different strategies for each cache size at a given bandwidth. The PSU strategy always provides the smallest values. The difference between the values of the PSU and the PML and PS strategies decreases with increasing bandwidths. The histogram for the cycle_3 pattern shows similar values for the cache sizes 4096 MB and 16384 MB at a given bandwidth, but more variation in the result values and greater differences between the strategies for a cache size of 512 MB. Taking a look at the sieve_3 pattern, similar values for the strategies can be recognized for each cache size at a given bandwidth. The relationships for the different strategies valid for all cache sizes and bandwidths regarding the WF_{start} metric can be found in Table 7.2. It indicates for all patterns, that the relationship PSU < PML and PSU < PS is valid for all cache sizes and a bandwidth of 5.76 Mbit/s. The relationship PSU < PML < PS is valid for all other cache size/bandwidth combinations. The multiple comparison test (see Appendix H, Table H.1) indicated that the PML and the PS strategies were significantly different for the sequence and the sieve_3 pattern at a bandwidth of 5.76 Mbit/s and cache sizes of 512 MB and 16384 MB and the cycle_3 pattern at a bandwidth of 5.76 Mbit/s and all cache sizes resulting in the relationship PSU < PML and PSU < PS. Comparing the relationships of the means of the single strategies for each pattern, bandwidth, and cache size, the result for bandwidth 5.76 Mbit/s and 4096 MB cache size was evaluated to PSU < PML < PS. While the observed differences are comparably low for bandwidth 5.76 Mbit/s and cache size 4096 MB for the sequence and the sieve_3 pattern, we weakened the statement for these cases to PSU < PML and PSU < PS.

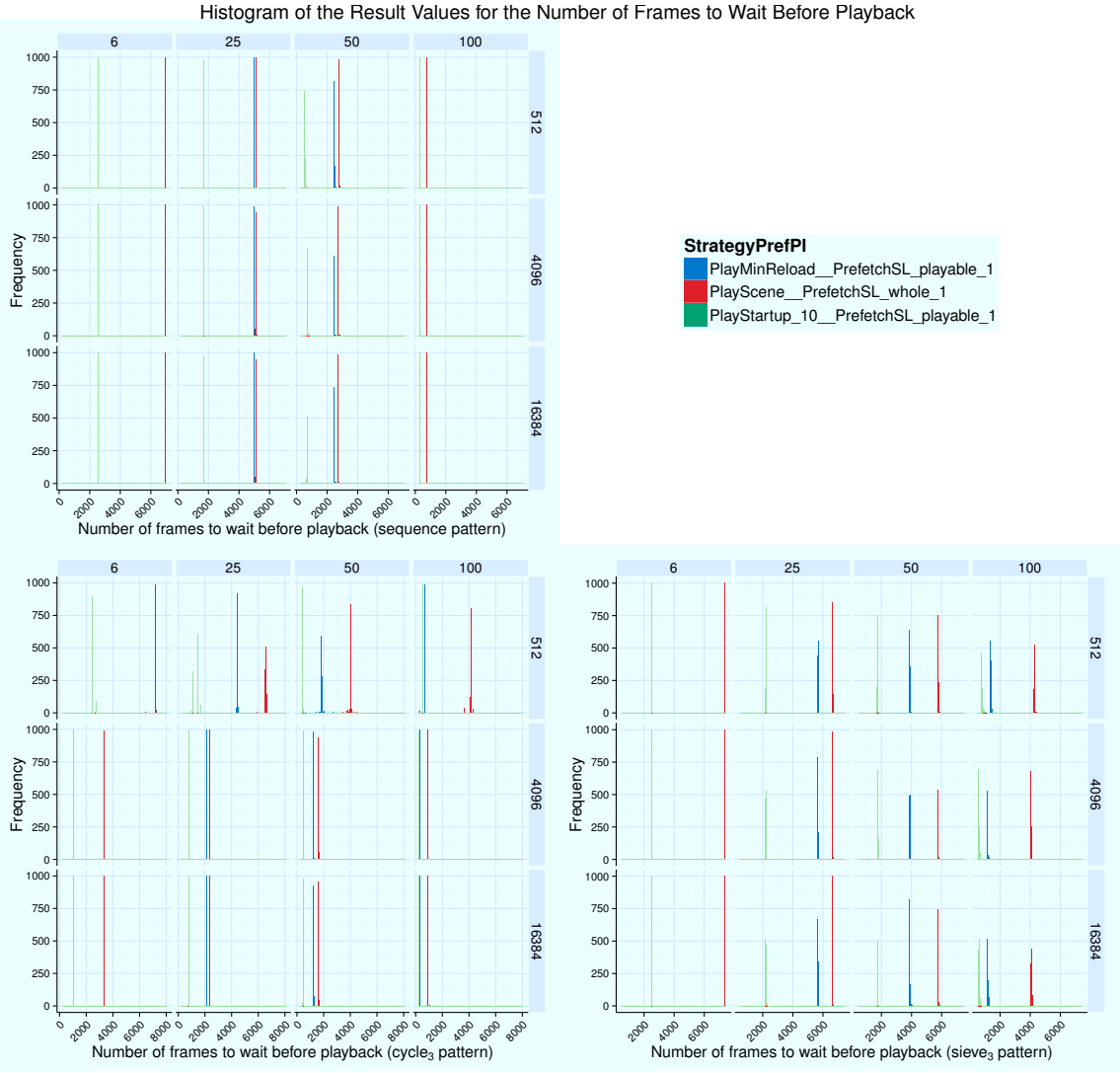


Figure 7.9.: Statistics - distribution of the number frames to wait before playback: sequence pattern (top left), $cycle_3$ pattern (bottom left), and $sieve_3$ pattern (bottom right); $binwidth$=50.

Cache size	Bandwidth			
	5.76 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
512 MB 4096 MB 16384 MB	PSU < PML ∧ PSU < PS	PSU < PML < PS		

Table 7.2.: Relationships between the strategies for the WF_{start} metric.

7.5.1.2. WT_{start} metric

The data distribution of the waiting time at the beginning of scenes WT_{start} indicates that the waiting times are independent from the cache sizes in the sequence pattern but vary from bandwidth to bandwidth (see Appendix I, Figure I.1). Thereby, all strategies provide

the same values for bandwidths of 50 Mbit/s and 100 Mbit/s and differ for the smaller tested bandwidths where the PML and the PS provide about the same but higher values than the PSU strategy. Taking a look at the $cycle_3$ pattern, it can be noted that the strategies have similar results for the cache sizes 4096 MB and 16384 MB at a given bandwidth but the values differ slightly from these for a cache size of 512 MB. The $sieve_3$ pattern again shows similar results for each cache size depending on the bandwidth. The PSU strategy provides smaller values for the bandwidths 5.76 Mbit/s, 25 Mbit/s, and 50 Mbit/s than both other tested strategies. The relationships for the different strategies valid for all cache sizes and bandwidths regarding the WT_{start} metric can be found in Table 7.3. For a bandwidth of 5.76 Mbit/s, the relationships PSU < PML and PSU < PS, for 25 Mbit/s and 50 Mbit/s PSU < PML < PS, and for 100 Mbit/s PSU < PS , PML < PS, and PML = PSU is valid. The multiple comparison test (see Appendix H, Table H.2) indicated that the PML and the PS strategies were not significantly different for all patterns for bandwidth 5.76 Mbit/s and all cache sizes. Furthermore, the PML and the PSU strategies were not significantly different for all patterns for bandwidth 100 Mbit/s and all cache sizes. More precisely, the PML and the PSU strategies had the same results for these test cases. All strategies were significantly different from each other for all patterns and all cache sizes for the bandwidths 25 Mbit/s and 50 Mbit/s. Comparing the relationships of the means of the single strategies for each pattern, bandwidth, and cache size the results for bandwidths 25 Mbit/s and 50 Mbit/s evaluated to PSU < PML < PS. Furthermore, the means of the single strategies for each pattern resulted in PSU < PS, PML < PS, and PML = PSU for each pattern for bandwidth 100 Mbit/s and all cache sizes. Accordingly, the findings for the comparison of the means did not contradict the multiple comparison test for all patterns, all cache sizes, and all bandwidths except 5.76 Mbit/s. Taking a look at the results for 5.76 Mbit/s bandwidth, the result for the means and the result for the multiple comparison test do not match for cache size 512 MB. While the comparison of the means evaluated to PSU < PS< PML, the multiple comparison test indicated no significant difference for this test case. We therefore unified the relationship of the strategies for all patterns and all cache sizes for bandwidth 5.76 Mbit/s to PSU < PML and PSU < PS.

Cache size	Bandwidth			
	5.76 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
512 MB 4096 MB 16384 MB	PSU < PML ∧ PSU < PS	PSU < PML < PS		PSU < PS ∧ PML < PS ∧ PML = PSU

Table 7.3.: Relationships between the strategies for the WT_{start} metric.

7.5.1.3. P_{sum} metric

Figure 7.10 shows the data distribution for the pauses during playback. All three patterns show about the same distribution of the values for cache sizes of 4096 MB and 16384 MB, whereby the values vary from bandwidth to bandwidth but do not differ from cache size to cache size. Some variations in the values can be noted for the $cycle_3$ pattern for a cache size of 512 MB. The data are overlapping completely or not at all for the number of pauses. The relationships for the different strategies valid for all cache sizes and bandwidths regarding the P_{sum} metric can be found in Table 7.4. The multiple comparison test (see Appendix H, Table

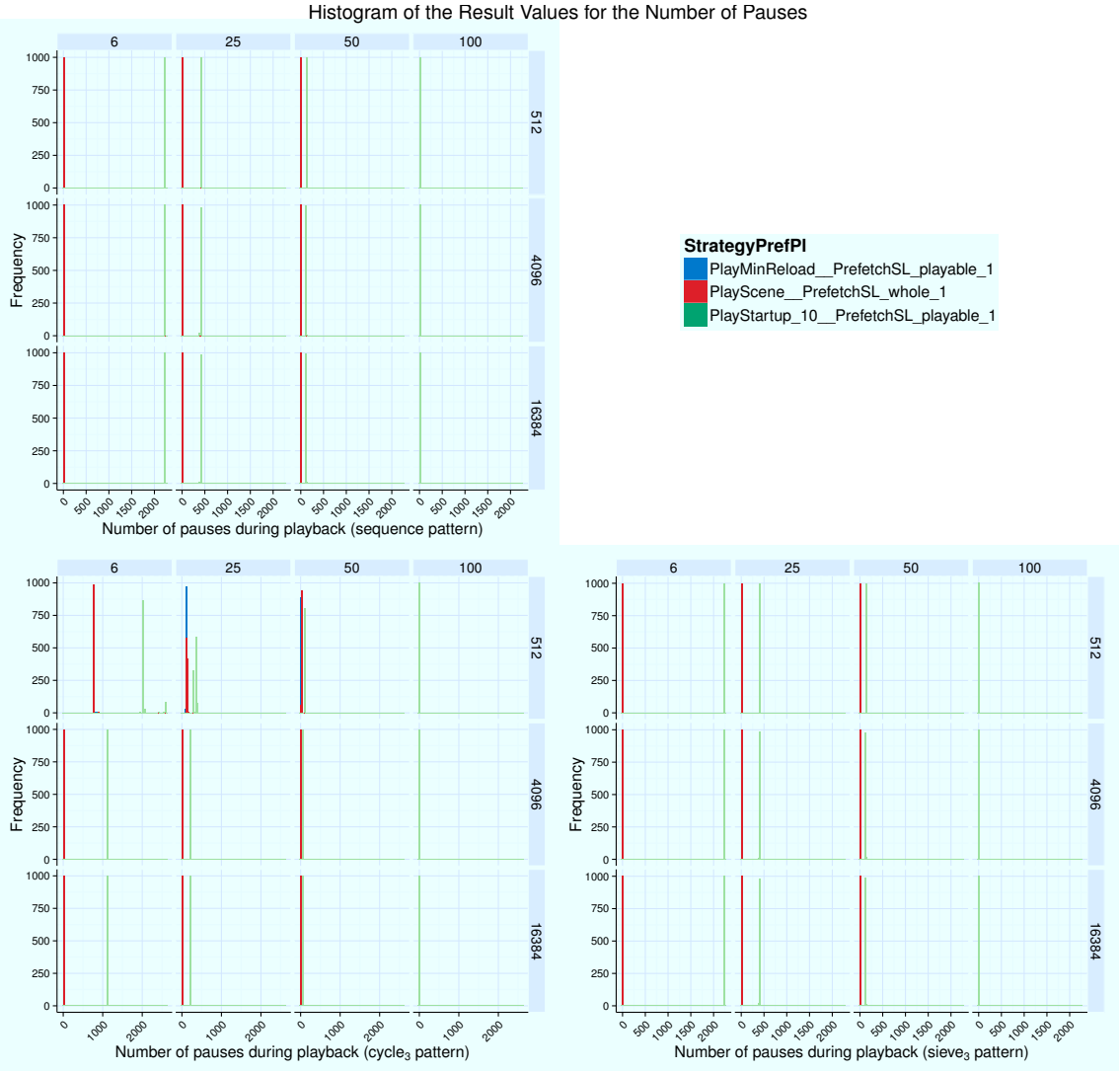


Figure 7.10.: Statistics - distribution of the number of pauses during playback: sequence pattern (top left), $cycle_3$ pattern (bottom left), and $sieve_3$ pattern (bottom right); $binwidth$=25.

H.3) was only conducted for the test cases with a cache size of 512 MB, because for larger cache sizes, the PML and the PS strategies had zero as result with a variance of zero and the results for the PSU strategy had significantly higher numbers of pauses with variances below 1.11 for all tests (see Appendix G, Table G.3). Furthermore the PSU strategy also had zero as result with a variance of zero for bandwidth 100 Mbit/s and cache sizes larger than 512 MB. The comparison of the means of all test cases with cache sizes larger than 512 MB lead to the relationships PS $<$ PSU, PML $<$ PSU, and PS $=$ PML for bandwidths smaller than 100 Mbit/s. For 100 Mbit/s bandwidth, the relationship PS $=$ PSU $=$ PML is valid. The multiple comparison test for the test cases with the $cycle_3$ pattern and cache sizes of 512 MB indicated significant differences for all cache size to bandwidth combinations except for the PML and the PS strategy tested at bandwidth 100 Mbit/s. These findings do not contradict the results of the comparison of the means which leads to the relationships PS $<$ PSU and PML $<$ PSU for all patterns, all bandwidths and a cache size of 512 MB.

Cache size	Bandwidth			
	5.76 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
512 MB	PS < PSU ∧ PML < PSU			
4096 MB 16384 MB	PS < PSU ∧ PML < PSU ∧ PS = PML			PML = PS = PSU

Table 7.4.: Relationships between the strategies for the P_{sum} metric.

7.5.1.4. *RDLV* metric

The data volume of repeated downloads *RDLV* shows the same values for the sequence pattern and the $sieve_3$ pattern as well as for the $cycle_3$ pattern with cache sizes of 4096 MB and 16384 MB, see Appendix I, Figure I.3. Thereby, the *RDLV* value is zero for all test runs with a standard deviation and a standard error of zero (see Appendix G, Table G.5). Accordingly, the following relationship between the strategies is valid: PML = PS = PSU, as listed in Table 7.5. Only for the $cycle_3$ pattern with a cache size of 512 MB, the values show a higher variance with overlappings in the distributions of the values. The corresponding figure can be found in Appendix I, Figure I.3. Taking a closer look at the $cycle_3$ pattern with cache size 512 MB and all bandwidths, the relationships between the means vary from bandwidth to bandwidth. For 5.76 Mbit/s, the relationships PSU < PML and PSU < PS are valid. For 25 Mbit/s, the relationship PML < PSU < PS and for 50 Mbit/s and 100 Mbit/s, the relationship PSU < PML < PS can be applied (see Appendix G, Table G.5 for detailed values). The multiple comparison test indicates that only the differences for the PML and the PS strategy are not significant for the $cycle_3$ pattern for bandwidth 5.76 Mbit/s and cache size 512 MB (see Appendix H, Table H.5).

Cache size	Bandwidth			
	5.76 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
512 MB	PS = PSU = PML ($sequence$ and $sieve_3$ pattern) (no universal valid statement for $cycle_3$ pattern)			
4096 MB 16384 MB	PS = PSU = PML			

Table 7.5.: Relationships between the strategies for the *RDLV* metric.

7.5.1.5. $DL_{not\ watched}$ and *DLV* metric

The results for the overall download volumes are shown in Figure 7.11. They are correlated to the results of the data volume of downloaded but not watched elements, which are therefore not discussed in detail here. The resulting values for the sequence pattern and the $cycle_3$ pattern for cache sizes of 4096 MB and 16384 MB are the same for all strategies, bandwidths, and cache sizes. They show no variance and the relationship PML = PS = PSU is valid (see Table 7.6 and Appendix G, Table G.4 and Table G.6). The results for the $sieve_3$ pattern differ therefrom. They vary only slightly from cache size to cache size, but show larger differences

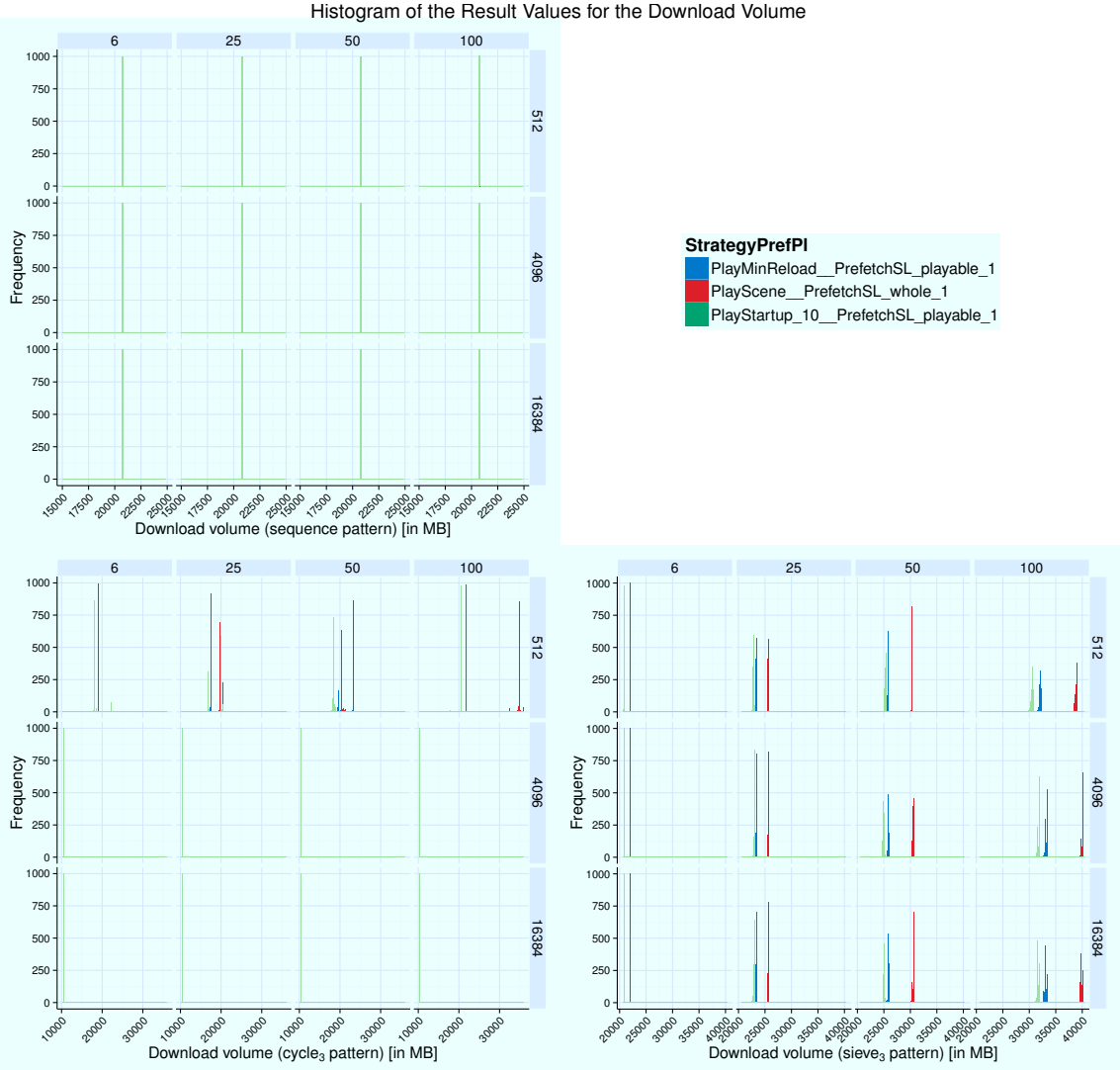


Figure 7.11.: Statistics - distribution of the download volumes: sequence pattern (top left), $cycle_3$ pattern (bottom left), and $sieve_3$ pattern (bottom right); $binwidth$=150.

between the different bandwidths. The PSU strategy always provides the lowest values. The PML and the PS strategies have higher values, whereby the difference between the results of the two strategies increases with increasing bandwidths. The variance is also slightly increasing with increasing bandwidth, but the distributions do not overlap because of the growing differences between the results.

For the $sieve_3$ pattern, the relationship PML < PS < PSU is valid for all bandwidths larger than 5.76 Mbit/s. For the bandwidth 5.76 Mbit/s the PML and the PS strategy do not show significant differences according to the multiple comparison (see Appendix H, Tables H.4 and H.6) test, but the means vary slightly. No commonly valid statement can be made for the strategies compared at cache size 512 MB and all bandwidths. As for the *RDLV* metric, for 5.76 Mbit/s, the relationships PSU < PML and PSU < PS are valid. For 25 Mbit/s, the relationship PML < PSU < PS and for 50 Mbit/s and 100 Mbit/s, the relationship PSU < PML < PS can be applied (see Appendix G, Table G.4 and Table G.6 for detailed values).

Cache size	Bandwidth			
	5.76 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
512 MB	PSU < PML ∧ PSU < PS ∨ (cycle$_3$ and sieve$_3$ pattern), PSU = PS = PML (sequence pattern)	PSU < PML < PS (sieve$_3$ pattern), PSU = PS = PML (sequence pattern), (no universal valid statement for cycle$_3$ pattern)		
4096 MB 16384 MB	PSU < PML ∧ PSU < PS (sieve$_3$ pattern), PSU = PS = PML (sequence and cycle$_3$ pattern)	PSU < PML < PS (sieve$_3$ pattern), PS = PSU = PML (sequence and cycle$_3$ pattern)		

Table 7.6.: Relationships between the strategies for the DLV metric.

7.5.1.6. Critical reflection

To summarize the findings of the tests for statistical significance, the following statements can be made:

- No commonly valid statement for all patterns, cache sizes, and bandwidths for all metrics can be made.
- Certain combinations of relationships are repeatedly valid:
  - PSU = PS = PML
  - PSU < PML ∧ PSU < PS
  - PSU < PML < PS
- Different relationships can be recognized for 5.76 Mbit/s or 100 Mbit/s, but the results for 25 Mbit/s and 50 Mbit/s are always the same.
- The standard deviations are zero or very small most of the time. Higher values for the standard deviation may occur for the cycle$_3$ pattern.

7.5.2. Evaluation of the Pre-fetch Strategies and Start Times

Due to the large number of possible combinations of start time, pre-fetch, and delete strategies we take a look at the pre-fetch strategies and different start times with a fixed delete strategy in this section. The delete strategies are evaluated separately in the next section then. We evaluate the pre-fetch and start strategies described in Sections 6.3 and 6.4.2 with the settings described in the Cartesian product in Equation 7.32 with the subsets defined in Equation 7.33. We analyze the results grouped by bandwidth and by cache size.

$$pattern \times probability \times duration \times anno \times size \times cache \times bw \times pbstart \times pref \times del \quad (7.32)$$

$$
\begin{aligned}
pattern &\in \{Cycle,\ Mirrorworld_3,\ Sieve_3,\ Split_3,\ Sequence\} \subseteq \mathcal{P}atterns \\
probability &\in \{prob_{best}(p_x),\ prob_{avg}(p_x),\ prob_{worst}(p_x)\} \subseteq \mathcal{P}robabilities \\
duration &\in \{dur_{short},\ dur_{medium},\ dur_{long},\ dur_{combi}\} \subseteq \mathcal{S}ceneDuration \\
anno &\in \{ac_{medium}\} \subseteq \mathcal{A}nnoCount \\
size &\in \{size_{low},\ size_{medium},\ size_{high}\} \subseteq \mathcal{S}izes \\
cache &\in \{512MB,\ 1024MB,\ 4096MB,\ 16384MB,\ 32768MB\} \subseteq \mathcal{C}acheSize \\
bw &\in \{5,76Mbit/s,\ 10Mbit/s,\ 16Mbit/s, \\
&\qquad 25Mbit/s,\ 32Mbit/s,\ 50Mbit/s,\ 100Mbit/s\} \subseteq \mathcal{B}andwidth \\
pbstart &\in \{PLAY\_SCENE,\ PLAY\_MIN\_REL(f_m), \\
&\qquad PLAY\_STARTUP(f_x)\} \subseteq \mathcal{P}laybackStart \\
&\qquad f_x \in \{125, 250, 375\} \\
pref &\in \{PREFETCH\_SL(z_{SL}, y, \Lambda, dist)\} \subseteq \mathcal{P}refetch \\
&\qquad z_{SL} = 1,\ y \in \{|p_x|, m, n\},\ \Lambda = 1,\ dist \in \{0, 1, 2, 3\} \\
del &= DELETE\_SD(\mu) \in \mathcal{D}elete,\ \mu = 1
\end{aligned}
\tag{7.33}
$$

7.5.2.1. Evaluation of the Start-up Frames (WF_{start})

This section tries to answer the following question: Which combination of algorithms/strategies results in the smallest number of start frames? Figure 7.12 shows the numbers of frames to wait for different bandwidths. A more detailed presentation of the results can be found in Appendix I, Figure I.4. The higher the bandwidth is, the lower are the numbers of frames to wait on average. It can be stated, that the number of frames to wait during start-up it the lowest for the strategies with a fixed start time, because the playback starts when a specified frame is available in the cache independent from available bandwidth, cache size, or possible waiting times during playback. The four curves representing the fixed start times have a downward tendency. This behavior is caused by the used pre-fetch strategy PREFETCH_SL with a pre-fetch depth of one scene. Using a pre-fetch depth of zero scenes, constant values as for the PLAY_SCENE setting with no pre-fetch would be achieved. The curves of the other strategies lie above those with a fixed start time. Thereby, three lines of each of the remaining start strategies with a pre-fetch depth higher than zero build a group. Only very little variations can be recognized for pre-fetch depths of one, two, or three scenes for the PLAY_MIN_REL and the PLAY_SCENE strategies. If no pre-fetch is used, PLAY_MIN_REL shows similar average results as PLAY_SCENE with pre-fetch. The curve is falling because with growing bandwidth, the amount of data which can be downloaded during playback is increasing. In contrast, the number of frames to wait is the same for each bandwidth using the PLAY_SCENE strategy with no pre-fetch, because the same number of frames has to be downloaded from zero on after each scene change. The PLAY_MIN_REL strategy has shorter average waiting times than the PLAY_SCENE strategy for each tested bandwidth. The PLAY_MIN_REL and the PLAY_SCENE strategies with a download depth of one scene have a slightly higher number of frames to wait at the beginning of the scene than those with a pre-fetch depth of more than one scene.

Grouping the data over the patterns gives hints on the results for scene graphs which contain a certain pattern more than others (Figure 7.12, center). Grouping over the used probabilities, statements for certain user behaviors can be made (Figure 7.12, right). Both, sequence and

$cycle_3$ pattern have the same order of the curves. The $sieve_3$ pattern in contrast shows a transposition of the PLAY_MIN_REL strategy with no pre-fetch and the group of PLAY_SCENE strategies with pre-fetch. This results from the higher number of scenes which have to be downloaded due to the branching patterns where each scene has three follow up scenes which are pre-fetched up to a certain point. The different probabilities of the paths also show this switch as a result which is caused by the very low amount of data which is downloaded for the $prob_{worst}$ path. As a result, fewer data for the path finally chosen by the viewer can be pre-fetched and the number of frames to wait gets larger as a consequence. No differences can be recognized between path probabilities for the strategies with no pre-fetch because the probabilities of future scenes have no influence thereby.

Figure 7.13 shows the numbers of frames to wait for different cache sizes. A more detailed presentation of the results can be found in Appendix I, Figure I.5. The average number of frames to wait is less dependent from the cache size than it is from the bandwidth. All categories of strategies show a slight downward tendency for the cache sizes from 512 to 4069 MB and a constant value for the cache sized from 4069 to 32768 MB. This behavior results from the fact, that all relevant data for one scene or the whole pattern fit into the three larger tested cache sizes. As described for the bandwidths, the PLAY_STARTUP strategies have the lowest number of frames to wait. The groups of PLAY_MIN_REL and PLAY_START strategies with a download depth of one scene or more are below those with no pre-fetch except for the smallest tested cache size of 512 MB. Taking a look at the values further grouped by pattern (Figure 7.13, center), it can be seen that the sequence and the $sieve_3$ pattern show about equal numbers of frames to wait for each cache size. The major difference is the position of the curves of the PLAY_MIN_REL strategy with no pre-fetch and the group of PLAY_START strategy with pre-fetch to each other. Thereby, the number of frames to wait with the PLAY_MIN_REL strategy with no pre-fetch is above the number of frames to wait with the PLAY_SCENE strategy in the sequence pattern and the other way around in the $sieve_3$ pattern. This behavior results from the high number of scenes to pre-fetch in the $sieve_3$ pattern which eliminates the advantage of pre-fetching due to the much higher data volume. The $cycle_3$ pattern shows the impact of the cache size in highly repetitive scene sequences. Small cache sizes require that scenes need to be repeatedly downloaded because they are deleted from playback to playback. The behavior results in a higher number of frames to wait at the beginning of a scene using small cache sizes. The selection of paths with a low probability results in a higher number of frames to wait for all cache sizes. The same behavior of the curves for the PLAY_MIN_REL strategy with no pre-fetch and the PLAY_SCENE strategies with pre-fetch can be recognized for the $prob_{worst}$ and the $prob_{best}$ settings. The low priority of the paths in the $prob_{worst}$ setting paired with a high number of scenes to pre-fetch results in larger waiting times than for the PLAY_MIN_REL strategy with no pre-fetch which is independent from any path probabilities.

Critical reflection: Regarding the frames to wait at the beginning of scenes, the following results can be summarized for the pre-fetch strategies and start times as evaluated in this section:

- The number of frames to wait is decreasing with increasing bandwidths for all strategies except for the PLAY_SCENE strategy without pre-fetch.
- The number of frame to wait is smaller for the tested PLAY_STARTUP strategies than for the PLAY_MIN_REL and the PLAY_SCENE strategies for each bandwidth.

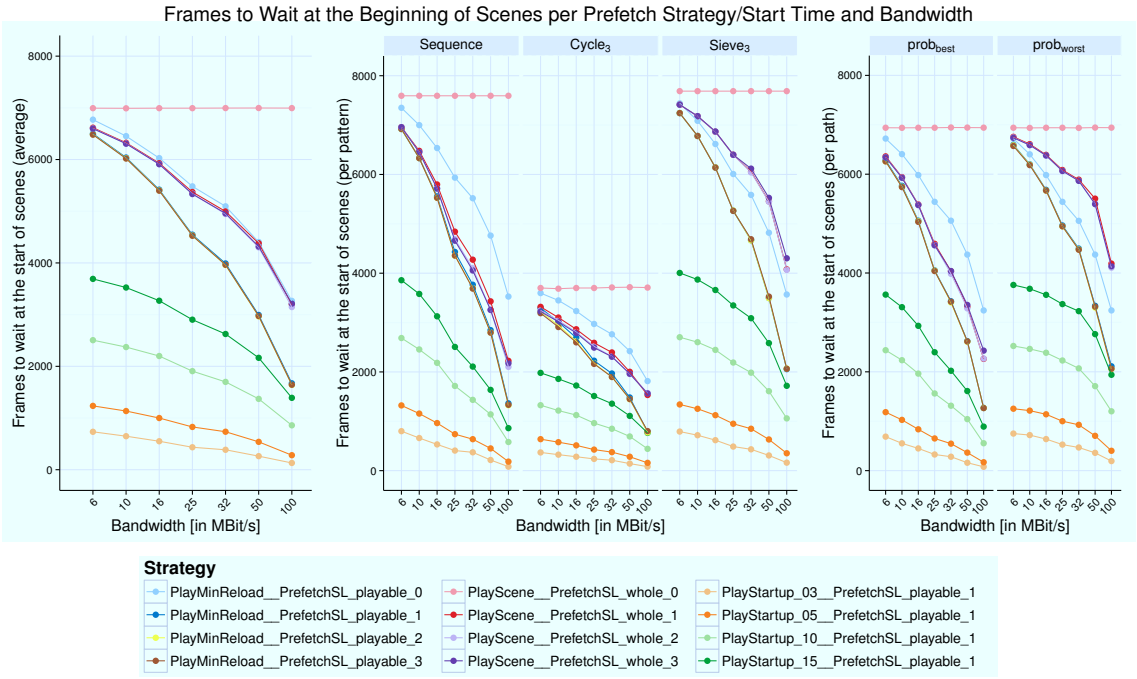


Figure 7.12.: Evaluation of the patterns (selected results) - frames to wait before playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by used probabilities (right).

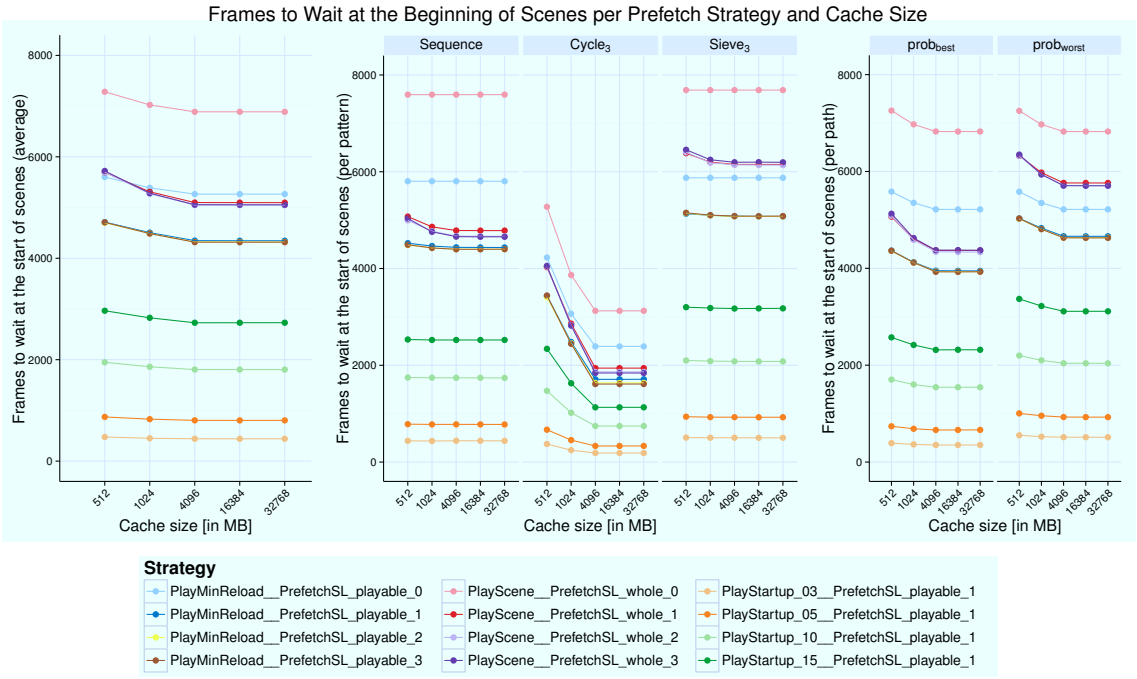


Figure 7.13.: Evaluation of the pre-fetch strategies (selected results) - frames to wait before playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by used probabilities (right).

- A pre-fetch depth of zero scenes results in higher numbers of frames to wait than pre-fetch depths of one scene or more. Thereby, the results for two or more pre-fetched scenes are not significantly better than for a pre-fetch of one scene.
- The cache size influences the number of frames to wait at the beginning of scenes if not all elements fit into the cache.

7.5.2.2. Evaluation of the Start-up Times WT_{start}

This section discusses the question: Which combination of algorithms/strategies results in the shortest start-up times? Figure 7.14 shows that the waiting time at the beginning of a scene is higher for the PLAY_MIN_REL and the PLAY_SCENE strategies than it is for the PLAY_STARTUP strategy for all bandwidths. The waiting time decreases with increasing bandwidths for all strategies. While the number of frames to wait showed significant differences between the PLAY_MIN_REL and the PLAY_SCENE strategies, the start times lie closer together due to the relation between the frame number to wait for and the increasing bandwidth. The results grouped by pattern (Figure 7.14, center) show no significant differences between each other. Only the $sieve_3$ pattern has slightly higher waiting times for the PLAY_SCENE strategies than for the other strategies. A more detailed presentation of the results can be found in Appendix I, Figure I.6.

Grouped by cache size (see Figure 7.15) an increase in waiting time can be recognized for all strategies without a fixed start time for increasing cache sizes. This behavior results from the circumstance that in cases with small cache sizes, the playback has to be started without all needed frames in the cache because the cache is full and the calculated frame which is needed for playback without wait cannot be downloaded. This is not necessary if the cache is large enough which leads to longer waiting times for larger caches.

Critical reflection: Taking a look at the waiting times at the beginning of scenes the following statements can be made for the pre-fetch strategies:

- The waiting times at the beginning of scenes are decreasing with increasing bandwidths.
- The waiting times are shorter for the tested PLAY_STARTUP strategies than for the PLAY_MIN_REL and the PLAY_SCENE strategies for each bandwidth except 100 Mbit/s.
- The waiting times for the PLAY_MIN_REL and the PLAY_SCENE strategies are very similar and vary slightly between patterns, but the waiting times for the PLAY_STARTUP strategies do not vary.
- The waiting times for the strategies without a fixed start time increase with increasing cache sizes using small cache sizes and stay constant for large cache sizes where all elements of a scene fit in.

7.5.2.3. Evaluation of the Pauses P_{sum}

While fixed start times result in a low or at least predictable waiting times at the beginning of scenes, they need more time for download during the scenes. This section now tries to answer the following question: Which combination of algorithms/strategies results in the fewest pauses during playback? Figure 7.16 shows the numbers of pauses during the playback

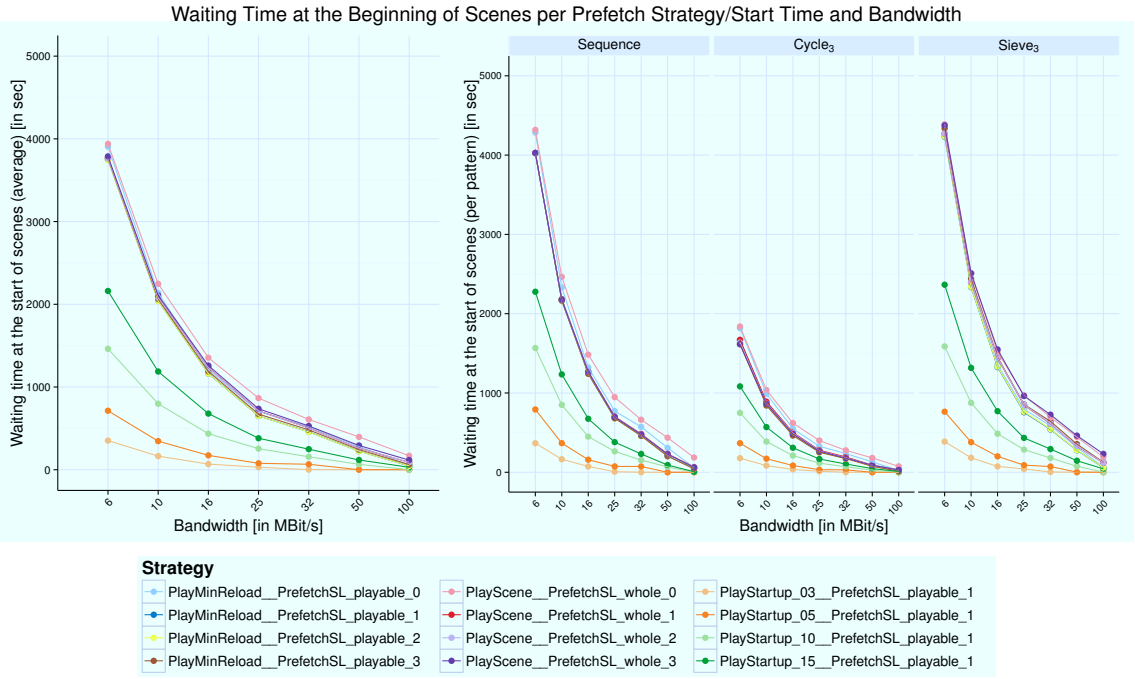


Figure 7.14.: Evaluation of the pre-fetch strategies (selected results) - waiting time before playback for different bandwidths: average for the whole test (left) and results grouped by pattern (right).

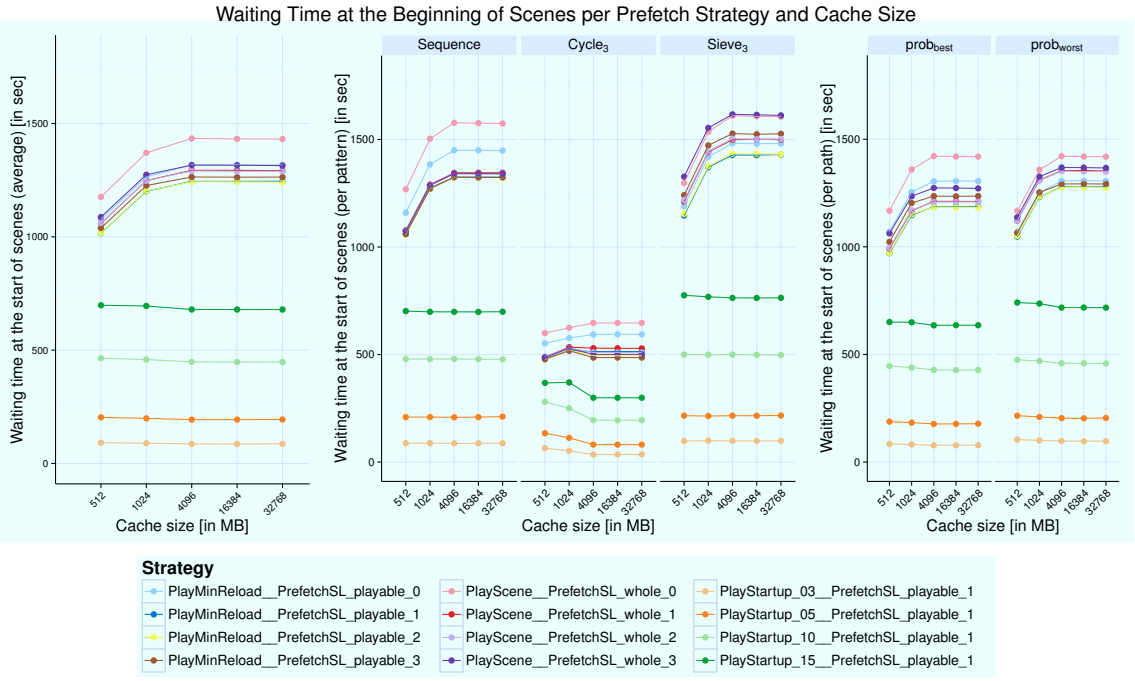


Figure 7.15.: Evaluation of the pre-fetch strategies (selected results) - waiting time before playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by used probabilities (right).

of scenes grouped by bandwidth. A more detailed presentation of these results can be found in Appendix I, Figure I.8.

Results for the number of frames to wait during a scene and the waiting times during a scene are not discussed in detail because they have a direct correlation the the number of pauses. At each pause, the player waits three seconds or 75 frames until the playback is started again. This may result in a large number of pauses and a jerky playback of the video. Another strategy would be to use longer waiting times and a achieve a smaller number of pauses as a result. The implemented behavior reduces the playback quality but gives continuous feedback to the viewer that the download is continued. Longer waiting times may result in the termination of the whole video because the user may assume that there might be a problem and playback will not be continued. Both strategies should be further investigated in future work.

The average number of pauses is decreasing with increasing bandwidths for all strategies. The PLAY_MIN_REL and the PLAY_SCENE strategies show significantly smaller numbers of pauses during scenes than the PLAY_STARTUP strategies. Thereby, the number of scenes which are pre-fetched does not matter. Grouped by pattern (Figure 7.16, center), the $cycle_3$ pattern has a lower number of pauses for the PLAY_STARTUP strategies than the other patterns. The PLAY_MIN_REL and the PLAY_SCENE strategies have higher numbers of pauses in the $cycle_3$ pattern in contrast. Grouped by cache size (Figure 7.16, right), the number of pauses are about the same for each cache size for the PLAY_STARTUP strategies. Considering the PLAY_MIN_REL and the PLAY_SCENE strategies, the pauses are decreasing with increasing cache sizes. No pauses are needed for cache sizes larger than 4096 MB. With smaller cache sizes, the number of pauses decreases with increasing bandwidth, because the needed elements are downloaded faster then, in case a scene did not fit into the cache.

Figure 7.17 (more detailed in Appendix I, Figure I.9) shows the numbers of pauses during the playback of scenes for different cache sizes. As described for the grouping by bandwidth, the results for the number of frames to wait during a scene and the waiting times during a scene for the different cache sizes have a direct correlation to the number of pauses as well. Therefore they are not discussed in detail here. The average number of pauses is decreasing with increasing cache sizes for all patterns. The number of pauses is higher for all cache sizes using the PLAY_STARTUP strategy than for the PLAY_MIN_REL and the PLAY_SCENE strategies. Thereby, the difference for the number of pauses for smaller and larger cache sizes is higher for the PLAY_MIN_REL and the PLAY_SCENE strategies than for the PLAY_STARTUP strategies. No pauses occur for the tested cache sizes of 4096 MB and above. Grouped by pattern (Figure 7.17, center), the $cycle_3$ pattern has a lower number of pauses for all PLAY_STARTUP strategies except the case of a waiting time of 15 seconds and 512 MB cache than the other patterns. The PLAY_MIN_REL and the PLAY_SCENE strategies have higher numbers of pauses for the cache sizes 512 MB and 2014 MB in the $cycle_3$ pattern than in the other patterns. This behavior results from the amount of data which can be kept in the cache until it is needed one more time. Grouped by bandwidth (Figure 7.17, right), large differences in the number of pauses can be recognized for the PLAY_MIN_REL and the PLAY_SCENE strategies compared to the PLAY_STARTUP strategies using small bandwidths. With increasing bandwidth, the differences between the curves become less distinctive because the smaller cache sizes are compensated by the higher bandwidths.

Critical reflection: Regarding the number of pauses during playback, the following statements are valid for the pre-fetch strategy/start time combinations:

- The number of pauses is decreasing with increasing bandwidths for all strategies.

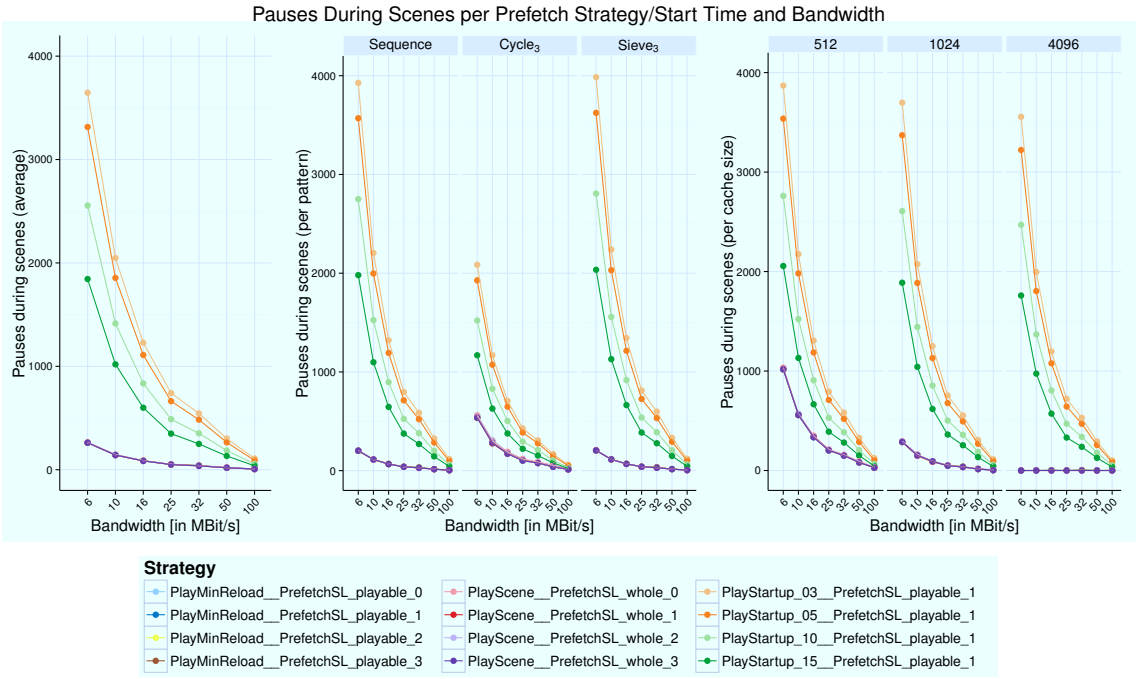


Figure 7.16.: Evaluation of the pre-fetch strategies (selected results) - pauses during playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

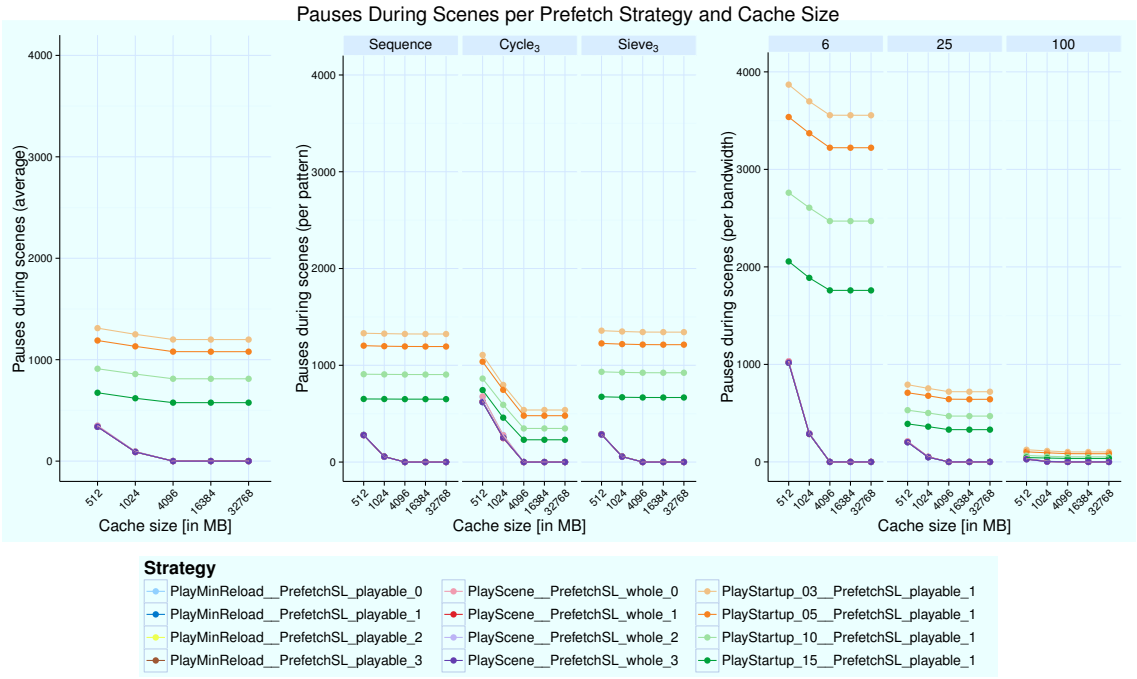


Figure 7.17.: Evaluation of the pre-fetch strategies (selected results) - pauses during playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

- Longer waiting times at the beginning of scenes result in smaller numbers of pauses during scenes and vice versa.
- The numbers of pauses for the PLAY_STARTUP strategies are significantly higher than the numbers of pauses for the PLAY_MIN_REL and the PLAY_SCENE strategies.
- Pauses can be avoided if the cache size is large enough to hold a whole scene up to the frame calculated as start frame.
- The difference between the PLAY_STARTUP strategies and the PLAY_MIN_REL and the PLAY_SCENE strategies is smaller in the cycle_3 pattern due to the repeated playback of elements.

7.5.2.4. Evaluation of the Download Volume of Elements not Watched $DL_{not\ watched}$

While waiting times at the beginning of scenes or during scenes are perceived directly by the viewers, the used download volume may have influence on available download rates in certain mobile phone contracts with a limited high speed download volume. The download volume of elements not watched should be kept as small as possible accordingly. Figure 7.18 shows the download volume of elements which are downloaded during the pre-fetch phases of the download, but not watched by the viewer for different bandwidths. The single strategies download elements in order to reduce the start-up time for a scene, but some elements are not watched, because the viewer choses another path at a selection. A more detailed overview can be found in Appendix I, Figure I.10. The average data volume of elements not watched decreases with decreasing bandwidths for all strategies which pre-fetch elements from future scenes. These findings result from the circumstance that the higher the bandwidth is the more elements of future scenes can be pre-fetched. Using a pattern width of three scenes which can be selected at each fork, the elements of two out of three scenes are not watched. Depending on the probability a path is selected with, the amount of data varies. The smaller the probability is, the more elements are scheduled at the end of a download queue and may not be downloaded using smaller bandwidths. When a comparably high bandwidth is used, even these elements can be downloaded into the cache. The more future elements are downloaded, the more will not be watched by the viewer. If no future elements are downloaded as in the PLAY_MIN_REL and the PLAY_SCENE strategies with no pre-fetch, no elements are downloaded into the cache which are not displayed. The download volume of elements not watched for these strategies is zero accordingly. Furthermore, the download volume of elements not watched is the highest for the PLAY_SCENE strategies with pre-fetch and low for PLAY_STARTUP with small start times. The download volume of elements not watched for the PLAY_MIN_REL strategies with pre-fetch and the PLAY_STARTUP strategies with higher start times lay between them and the curves intersect from bandwidth to bandwidth. Grouped by pattern (Figure 7.18, center), the amount of downloaded but elements not watched elements is very small for the sequence and the cycle_3 pattern. This results from the structure of the patterns. All elements are watched in the linear sequence pattern. Only for small cache sizes and high bandwidths, elements of future scenes may be deleted before they are viewed and then be downloaded again. This explains the increase of the curve from 50 Mbit/s to 100 Mbit/s in the sequence pattern. A small amount of downloaded but not watched elements can also be recognized with increasing bandwidths in the cycle_3 pattern. This pattern contains mainly scenes in a linear order, but also one fork. The pre-fetch at the fork results in the increasing download volume of elements not watched with increasing bandwidths. The

download volume increases significantly in patterns with many forks as illustrated for the $sieve_3$ pattern. Grouped by cache size (Figure 7.18, right), the amount of downloaded but not watched elements is slightly higher at small cache sizes than at higher ones. This can be explained by the fact that not all needed elements may fit in the cache and future elements have to be deleted and then downloaded again.

The data volume of elements not watched for different cache sizes can be found in Figure 7.19 and in more detail in Appendix I, Figure I.11. The average download volume of elements not watched is the highest for the PLAY_SCENE strategies with pre-fetch whereby it is the lower for a pre-fetch depth of one scene and higher for two or more scenes. It is lower for the PLAY_MIN_REL strategies with pre-fetch than for the PLAY_SCENE strategies with pre-fetch. Thereby it is lower for a pre-fetch depth of one scene and higher for a pre-fetch of more scenes. For both PLAY_MIN_REL and PLAY_SCENE strategies, the curves show a slightly increasing course. The PLAY_STARTUP strategies all show a slightly increasing course of the curves whereby the PLAY_STARTUP strategy with a start time of 15 seconds lies above the PLAY_MIN_REL strategy with a pre-fetch depth of one scene, all others lie below. Grouped by pattern (Figure 7.19, center), all strategies except those with no pre-fetch show decreasing curves for mainly linear sequences of scenes. Thereby the amount of downloaded but not watched elements decreases with increasing cache sizes because all elements that are downloaded will be watched eventually during the course of the video. No elements have to be deleted and reloaded due to the large enough cache. Taking a look at the $sieve_3$ (and the other patterns with forks), it can be recognized that all strategies, especially the PLAY_MIN_REL and the PLAY_SCENE strategies with pre-fetch, have an increasing curve for increasing cache sizes. This behavior results from the fact that the download cannot pre-fetch as many elements with smaller cache sizes than with larger ones. The download has to stop when the cache is full and no further elements can be downloaded which results in the increasing number of elements that were downloaded but not watched with increasing cache sizes. Grouped by bandwidth (Figure 7.19, right), it can be noted that the amount of downloaded but not watched data is the smaller the lower the bandwidth is. The download volume of elements not watched is very high for the PLAY_SCENE strategies with pre-fetch.

Critical reflection: Taking a look at the results for the data volume of downloaded but not watched elements, the following statements can be made for the pre-fetch strategies:

- The data volume of downloaded but not watched elements increases with increasing bandwidths for all strategies except those without pre-fetch.
- The data volume of downloaded but not watched elements strongly depends on the structure of the pattern. Patterns with many forks result in higher data volumes of downloaded but not watched elements than patterns with few forks.
- The results stay constant if the cache size is large enough. For small cache sizes, the curves either increase or decrease depending on the used pattern.

7.5.2.5. Evaluation of the Download Volume of Repeatedly Downloaded Elements *RDLV*

Besides the download volume of not watched elements, the download volume of repeatedly downloaded elements should be kept as small as possible. Figure 7.20 shows the download volume of the repeatedly downloaded elements grouped by cache size. A more detailed overview over this part of the evaluation can be found in Appendix I, Figure I.12. Except for

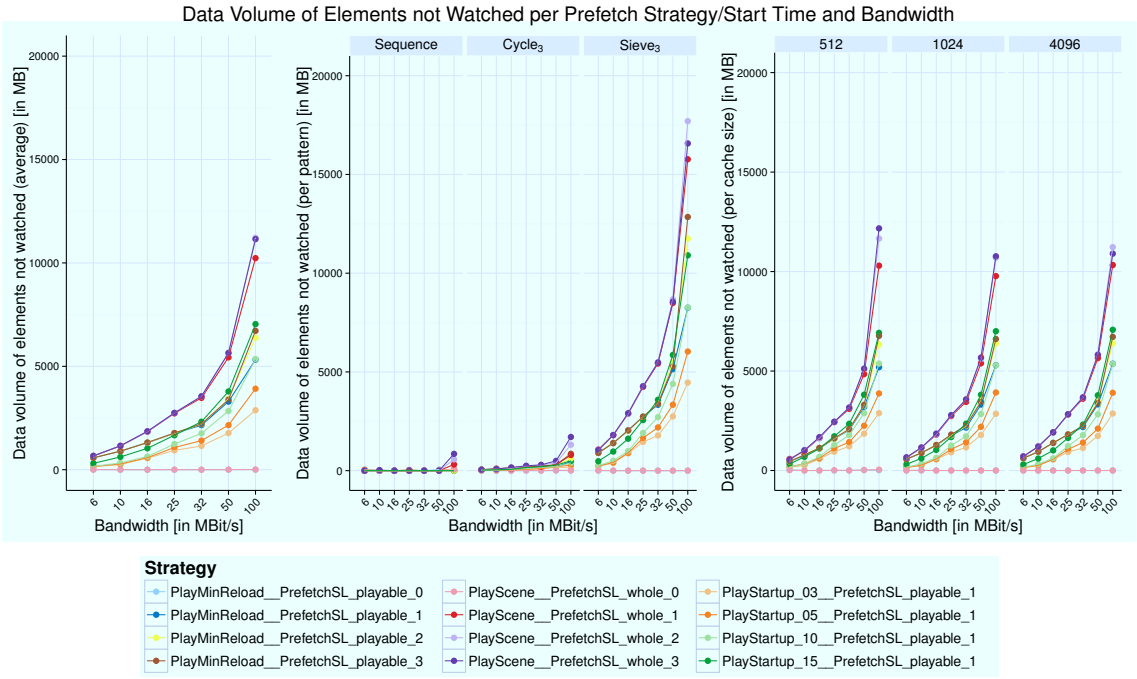


Figure 7.18.: Evaluation of the pre-fetch strategies (selected results) - data volume of elements not watched for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

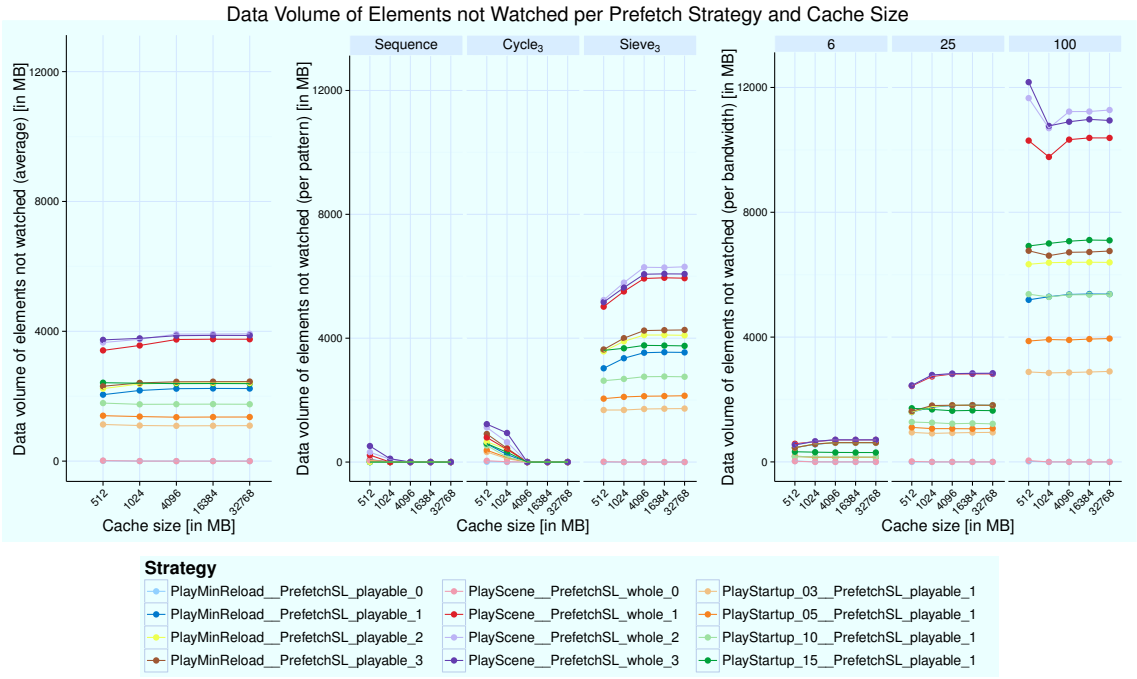


Figure 7.19.: Evaluation of the pre-fetch strategies (selected results) - data volume of elements not watched for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

the bandwidth of 100 Mbit/s, the average data volume of repeatedly downloaded elements of the regarded strategies varies from bandwidth to bandwidth and has no clear tendency. This behavior results mainly from the results achieved for the $cycle_3$ pattern as can be seen in the center of Figure 7.20. The data volume is comparably low for all patterns without a cycle and only increases for a bandwidth of 100 Mbit/s where elements are pre-fetched, which may be deleted due to small cache sizes. The large variations in the $cycle_3$ pattern result from the repeated viewing of certain scenes in addition to small cache sizes, where all elements of a cycle do not fit into the cache. Depending on the cache size and the download volume a certain amount of data has to be deleted from the cache and then downloaded again for another round in the cycle. Figure 7.20 (right) illustrates this behavior in more detail.

The download volume of repeatedly downloaded elements for the cache sizes shows a clear falling tendency for small caches and consistently low values for larger cache sizes (see Figure 7.21, in more detail Appendix I, Figure I.13). The average data volume of repeatedly downloaded elements is larger for the PLAY_SCENE strategies for smaller cache sizes while it shows only small variations for the other strategies. As Figure 7.21 (center) shows, the increased values in the diagram with the average values results mainly from the values of the $cycle_3$ pattern. If the cache size is large enough, no elements need to be downloaded repeatedly, because none are deleted from the cache during playback. The combination of small cache sizes with high bandwidths results in a higher volume of repeatedly downloaded elements (Figure 7.21, right), because the high bandwidth enables downloads of future elements. These elements may be deleted due to the necessity to download other elements needed more recently while using small cache sizes. Accordingly they have to be downloaded again at a later time.

Critical reflection: The analysis of the data volume of repeatedly downloaded elements leads to the following conclusions for the pre-fetch strategies:

- The data volume of repeatedly downloaded element is very low for patterns without cycles, the only recognizable increase for the $cycle_3$ pattern can be seen for a bandwidth of 100 Mbit/s where elements from the pre-fetch at the fork may have to be deleted in order to be able to play a scene.
- The data volume of repeatedly downloaded elements is varying around a certain value in the $cycle_3$ pattern without a clear increase or decrease for bandwidths of 50 Mbit/s and smaller. The values differ more for 100 Mbit/s. This results form different points in time for delete due to small cache sizes and slight variations in the used internal timing model of the player.
- The data volume of repeatedly downloaded elements is increasing for cache sizes smaller than 4096 MB and is constantly zero for cache sizes of 4096 MB and above.

7.5.2.6. Evaluation of the Download Volume *DLV*

The overall download volume depends on the underlying data volume (resulting from the underlying pattern and the sizes of the elements), the repeatedly downloaded elements, and the downloaded but not watched elements. This section deals with the question: Which combination of algorithms/strategies results in the smallest download volume? Figure 7.22 (for more details see Appendix I, Figure I.14) shows the average download volume for the different bandwidths. Except for the strategies with no pre-fetch, all curves show an increasing development. The highest data volume is caused by the PLAY_SCENE strategies with pre-

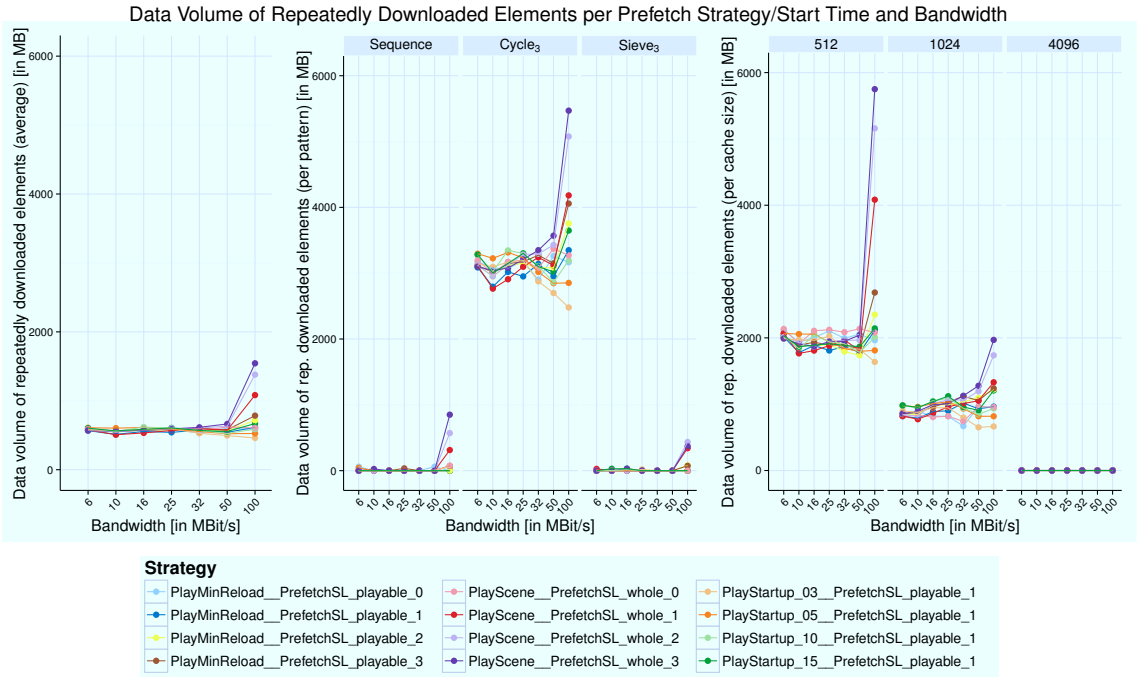


Figure 7.20.: Evaluation of the pre-fetch strategies (selected results) - data volume of repeatedly downloaded elements for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

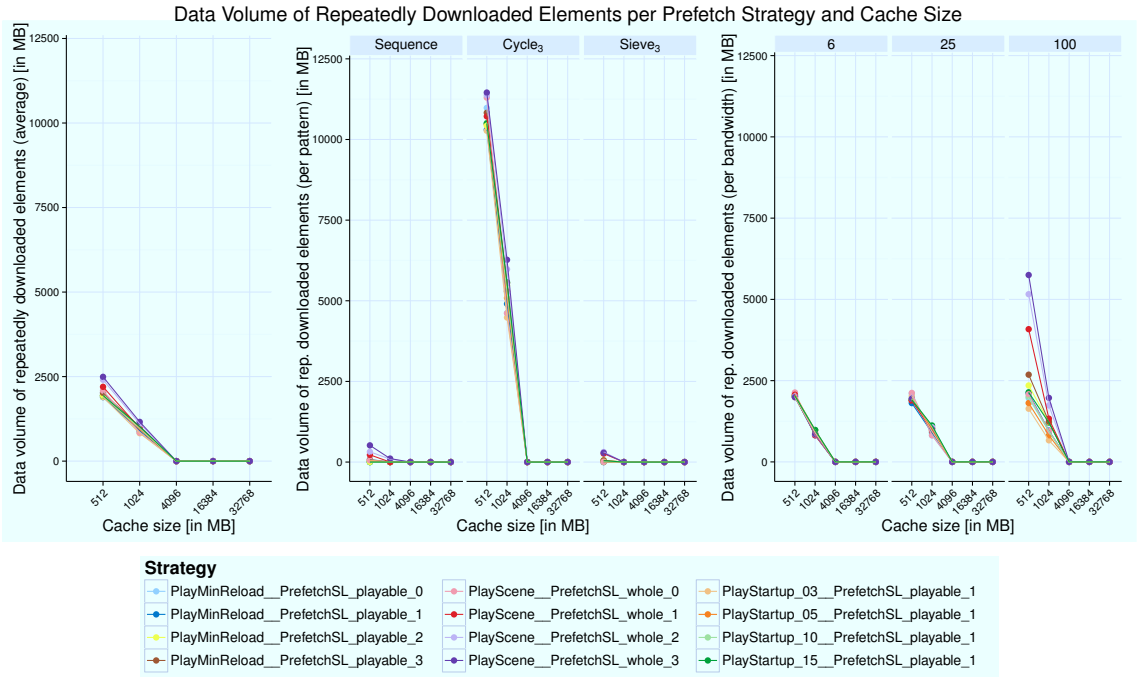


Figure 7.21.: Evaluation of the pre-fetch strategies (selected results) - data volume of repeatedly downloaded elements for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

fetch followed by the PLAY_MIN_REL strategies with pre-fetch and the PLAY_STARTUP with a start time of 15 seconds again followed by the PLAY_STARTUP strategies with smaller start times. The download volume is constant for strategies with no pre-fetch, because thereby only these elements are downloaded which are needed without any overhead. Grouped by the pattern (Figure 7.22, center), all strategies have about the same download volume for the sequence pattern. The smallest download volume can be achieved for the $cycle_3$ pattern with its high number of repeated views of each scene. The more forks a pattern has, the higher is the download volume for increasing bandwidths. Taking a look at the different cache sizes (Figure 7.22, right), it can be adhered that the download volume is slightly decreasing with increasing cache sizes while the differences between the results of the different strategies remain about the same.

The download volume is decreasing for increasing cache sizes (see Figure 7.22, for more details see Appendix I, Figure I.15). The average download volume is the highest for the PLAY_SCENE strategies with pre-fetch and the lowest for all strategies without pre-fetch for all cache sizes. Grouped by pattern (Figure 7.22, center left), all strategies have about the same download volume for the sequence pattern. Taking a look at the $cycle_3$ pattern, decreasing download volumes can be seen for small cache sizes while the smaller download volumes appear at larger cache sizes. The $sieve_3$ pattern in contrast shows slightly increasing curves for the PLAY_MIN_REL and the PLAY_SCENE strategies with pre-fetch and constant curves for all other strategies. Regarding the assigned path probabilities (Figure 7.22, center right), it can be noted that the download volume is higher if the user selects the paths with low probabilities than with high probabilities. This results from the fact, that more data are downloaded from higher prioritized paths which are not watched then, while the download of data from lower prioritized paths need to be downloaded for playback. Accordingly, a higher amount of data is discarded which results in a higher download volume. Grouped by bandwidth (Figure 7.22, right), the diagram shows that the download volume varies only very little between the different strategies for smaller bandwidth, while the differences are larger for higher bandwidths. The reason for this result is a higher amount of data that can be downloaded during the pre-fetch.

Critical reflection: Taking a look at the overall download volume, the following findings can be recorded for the pre-fetch strategies:

- The download volume is increasing with increasing bandwidths (because a further pre-fetch is possible).
- The download volume is decreasing for small cache sizes and stays constant if the scenes fit into the cache up to the frame from which on the scene can be played without pauses.
- The download for patterns with cycles is smaller than for the other ones because once downloaded elements are viewed more than once.
- The download volume strongly depends on the bandwidth but only little from the cache size.

7.5.3. Evaluation of the Delete Strategies

As in the evaluation of pre-fetch strategies and start times, we evaluate the delete strategies described in Section 6.5 with the settings described in the Cartesian product in Equation 7.32

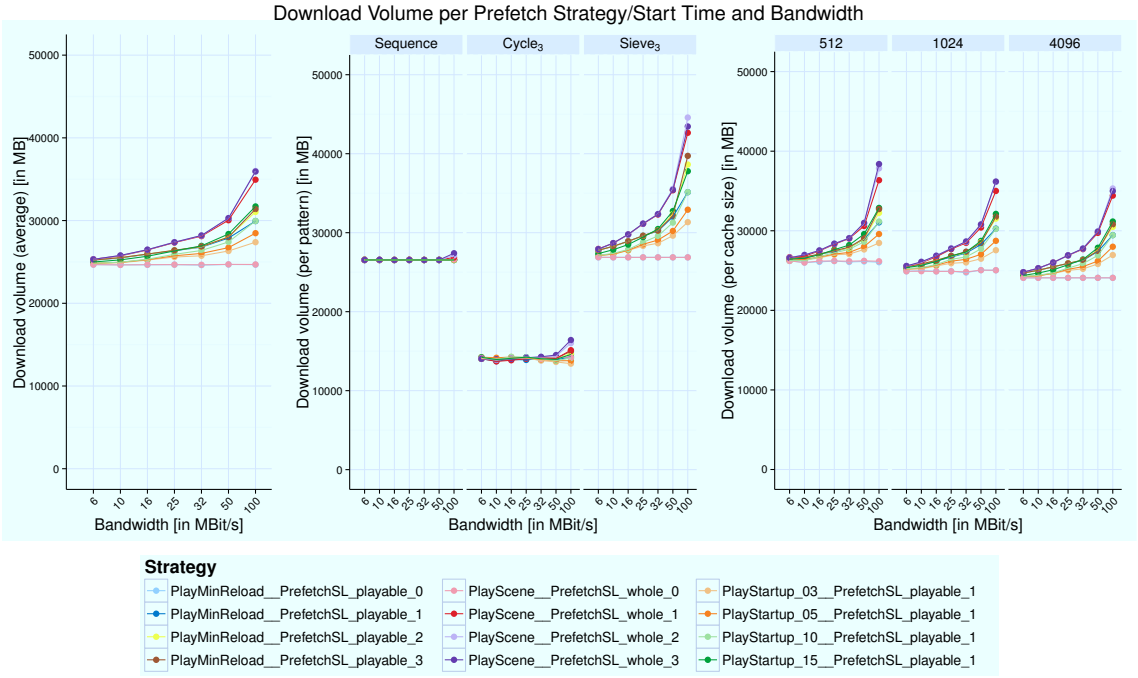


Figure 7.22.: Evaluation of the pre-fetch strategies (selected results) - download volume for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

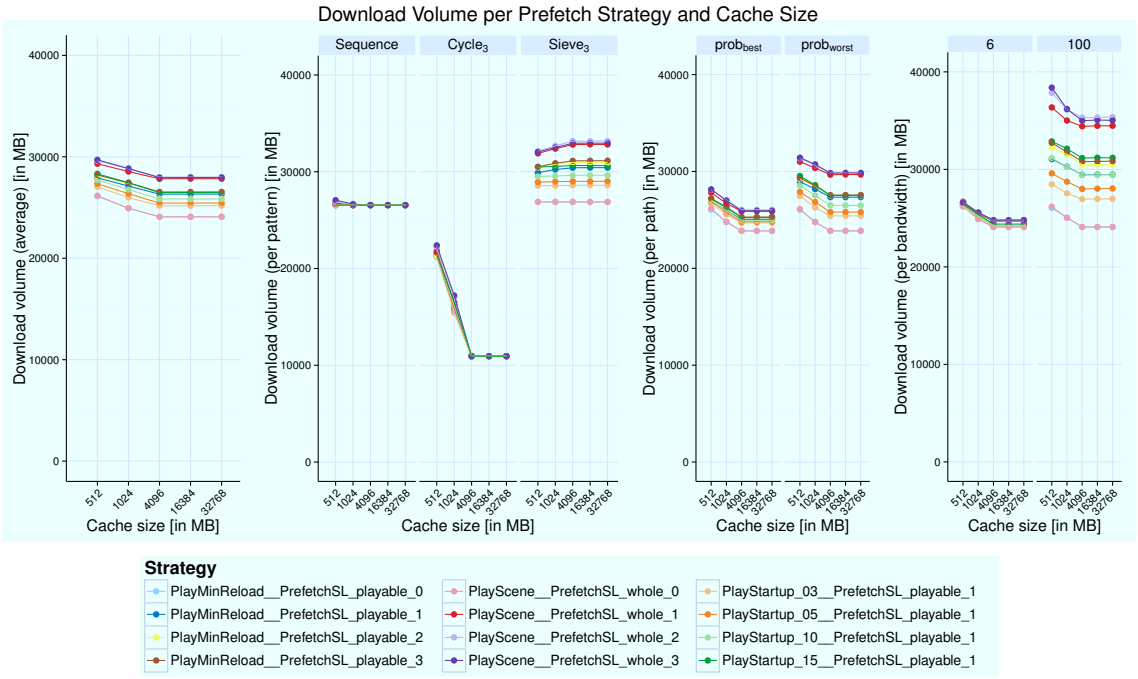


Figure 7.23.: Evaluation of the pre-fetch strategies (selected results) - download volume for different cache sizes: average for the whole test (left), results grouped by pattern (center left), results grouped by used probabilities (center right), and results grouped by bandwidth (right).

with the subsets defined in Equation 7.33. We also analyze the results for different bandwidths and cache sizes. We furthermore evaluate each delete strategy with different settings for the amount of data that is deleted during on delete operation, namely 10 % and 20 % of the cache size.

7.5.3.1. Evaluation of the Start-up Frames (WF_{start})

Figure 7.24 shows the average number of frames to wait at the beginning of a scene for different bandwidths. A more detailed overview over the results can be found in Appendix I, Figure I.16. The higher the bandwidth is, the lower is the number of frames to wait for each delete strategy. Thereby, the curve of the DELETE_LRU strategies is slightly above the curve of the other strategies. Grouped by pattern (see Figure 7.24, center), it can be noted that the difference between the strategies is the highest for the cycle_3 pattern. All other patterns show very little differences for small bandwidths and little difference for higher bandwidths. Grouped by cache size (Figure 7.24, center), differences can be noticed for cache sizes of 512 MB and 1024 MB between the DELETE_LRU strategies and all other strategies. For all higher cache sizes, all strategies show nearly the same results, because only very few delete operations are processed. The results for the start-up times at the beginning of scenes show very similar results and are not discussed here in more detail. An overview over the results can be found in Appendix I, Figure I.18.

Figure 7.25 (and in more detail in Appendix I, Figure I.17) shows the average number of frames to wait at the beginning of scenes for different cache sizes. While the results for the DELETE_LRU strategies are higher for small cache sizes than for all other strategies, the results are about the same for larger cache sizes. This relation of curves of the DELETE_LRU strategies to the other strategies can also be found if the results are grouped by pattern or by bandwidth. Grouped by pattern (Figure 7.25, center), only the cycle_3 pattern shows a deviation from the other patterns. Thereby, all curves have a decreasing tendency for small cache sizes. Grouped by bandwidth a falling tendency can be noticed from bandwidth to bandwidth. Thereby, the differences between the DELETE_LRU strategies and the other strategies is growing with increasing bandwidths. The results for the start-up frames at the beginning of scenes show according results which are not discussed here in more detail. An overview over the results can be found in Appendix I, Figure I.19.

Critical reflection: The evaluations of the number of the start frames for the different delete strategies revealed the following findings:

- The number of frames to wait and the waiting times at the beginning of scenes are decreasing for all strategies with increasing bandwidths.
- The results for the DELETE_SD and the DELETE_DPROB strategies are very similar independent from the selected μ or the amount of data to delete if the cache is full. The DELETE_LRU strategy shows slightly higher numbers of frames to wait especially for the cycle_3 pattern where the underlying structure is very important for delete decisions.
- The results for the DELETE_LRU strategies are worse than the others in all patterns where small cache sizes are used.

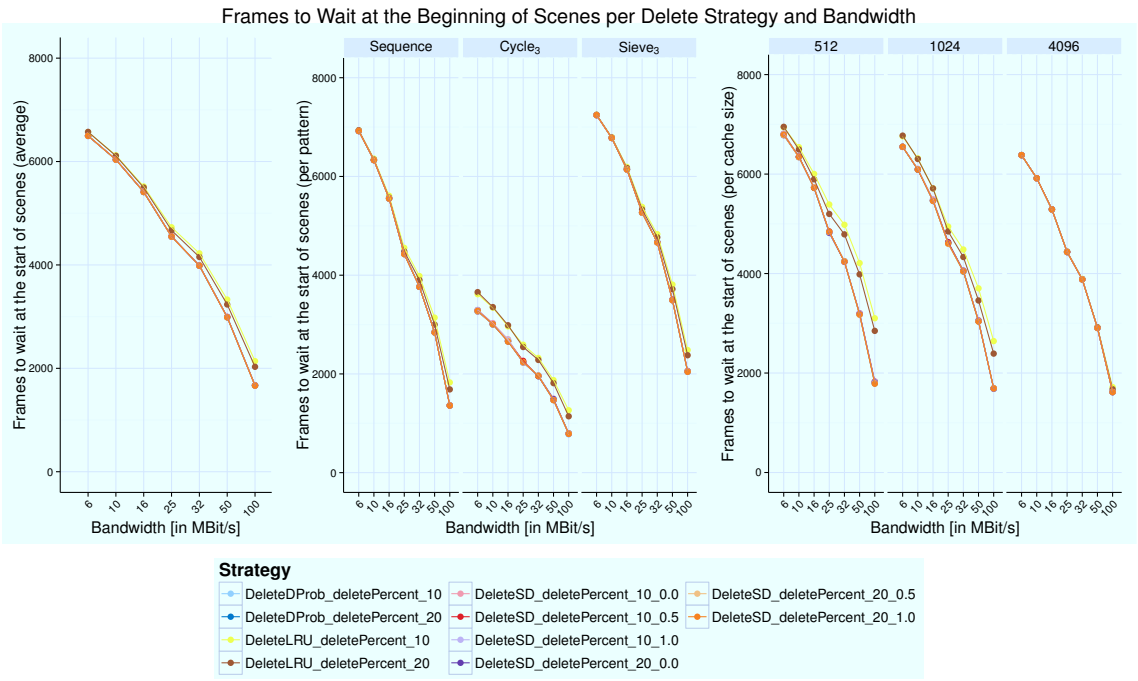


Figure 7.24.: Evaluation of the delete strategies (selected results) - frames to wait before playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

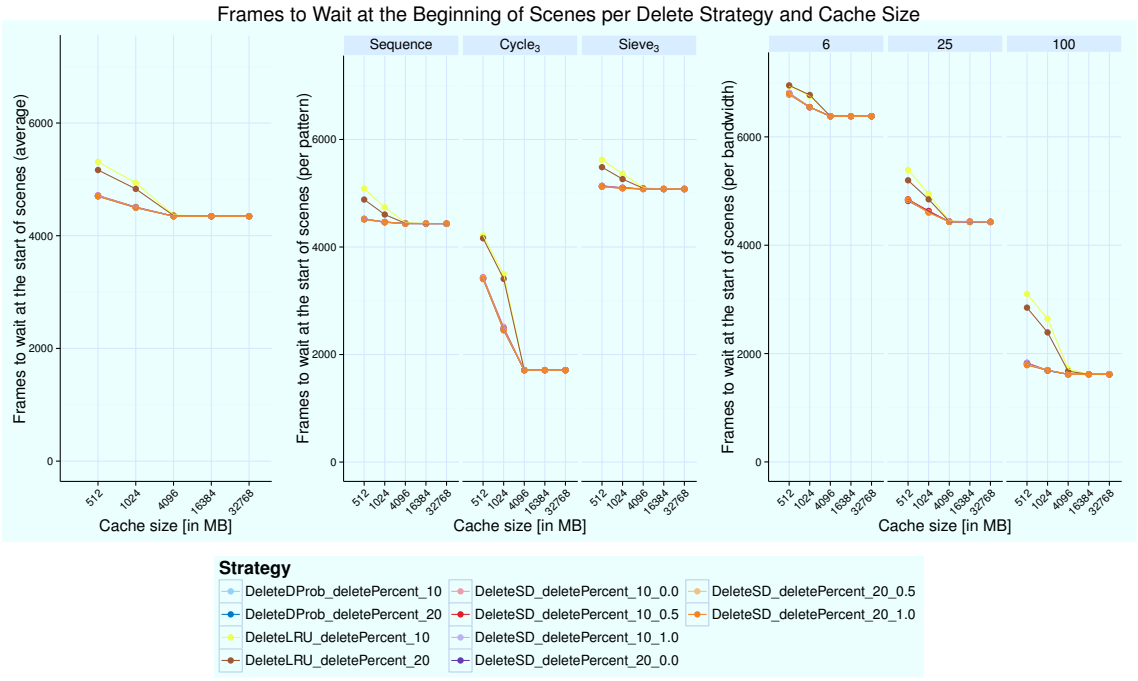


Figure 7.25.: Evaluation of the delete strategies (selected results) - frames to wait before playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

7.5.3.2. Evaluation of the Pauses (P_{sum})

Trying to find an answer to the question "Which combination of algorithms/strategies results in the fewest pauses during playback?", we evaluate the pauses for different bandwidths and cache sizes. Figure 7.26 shows the average number of pauses during scenes grouped by bandwidth. Thereby, the DELETE_LRU strategies show slightly better results than the other strategies for all bandwidths, but the differences decrease with increasing bandwidths. Grouped by pattern (see Figure 7.26, center), only the $cycle_3$ pattern shows varying results for the DELETE_LRU strategies compared to the other strategies. Thereby the DELETE_LRU strategies show slightly better results than the other strategies. This behavior is caused by the behavior of the strategies dealing with small cache sizes (see Figure 7.26, right). The larger the cache sizes get, the lesser elements need to be deleted and the lesser effects can be recognized for the number of pauses. These results are directly correlated with the numbers of frames to wait during the scenes and the waiting times during scenes for the different bandwidths which are not described here in more detail.

Figure 7.27 shows the average number of pauses during scenes for different cache sizes. A detailed overview can be found in Appendix I, Figure I.21. The pauses between scenes decrease with increasing cache sizes for all delete strategies. Thereby, the values for the DELETE_LRU strategies are slightly better than those for the other strategies. The reason for the course of the curve can be found in the results of the strategies for the $cycle_3$ pattern for small cache sizes (see Figure 7.27 (center) and Figure 7.27, right). As for the results for the bandwidths, the results of the number of frames to wait at the beginning of scenes and the waiting time at the beginning of scenes for the cache sizes are correlated to the number of pauses as well.

Critical reflection: Taking a look at the number of pauses during scenes for all tested delete strategies, the following statements can be made:

- The DELETE_LRU strategies show slightly better results for the number of pauses than the other strategies, especially for the $cycle_3$ pattern.
- No differences between the strategies can be seen for cache sizes of 4096 MB or more, because no delete has to be performed during the video then.

7.5.3.3. Evaluation of the Download Volume *DLV*

Trying to answer the question "Which combination of algorithms/strategies results in the smallest download volume?", we analyze the download volume of elements not watched, the data volume of repeatedly downloaded elements, as well as the overall download volume. The data volume of elements not watched is increasing with growing bandwidths (see Appendix I, Figure I.22). Thereby, the download volume is always higher for the DELETE_LRU strategies than for the other strategies, especially for small cache sizes (see Appendix I, Figure I.23). The data volume of repeatedly downloaded elements shows only little variance for all delete strategies except for the DELETE_LRU strategies where it is increasing with growing bandwidths (see Appendix I, Figure I.24). This behavior strongly relates to the results of the strategies in the $cycle_3$ pattern dealing with small cache sizes. The same tendencies can be found for the DELETE_LRU strategies grouping the results by cache size. Thereby, the

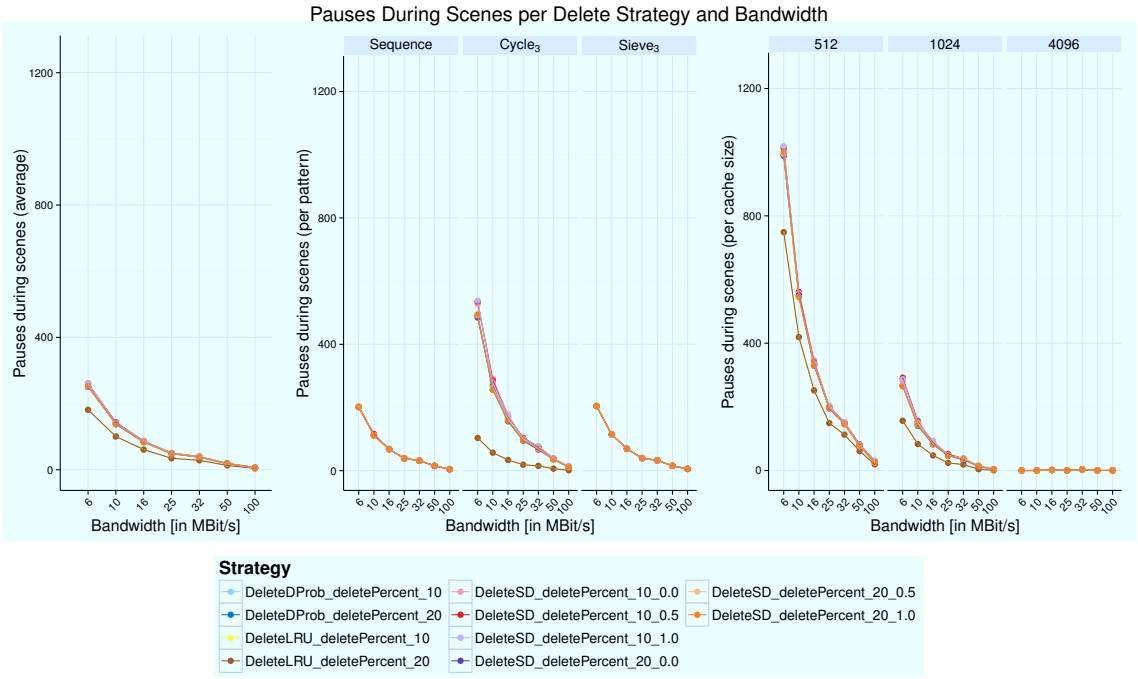


Figure 7.26.: Evaluation of the delete strategies (selected results) - pauses during playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

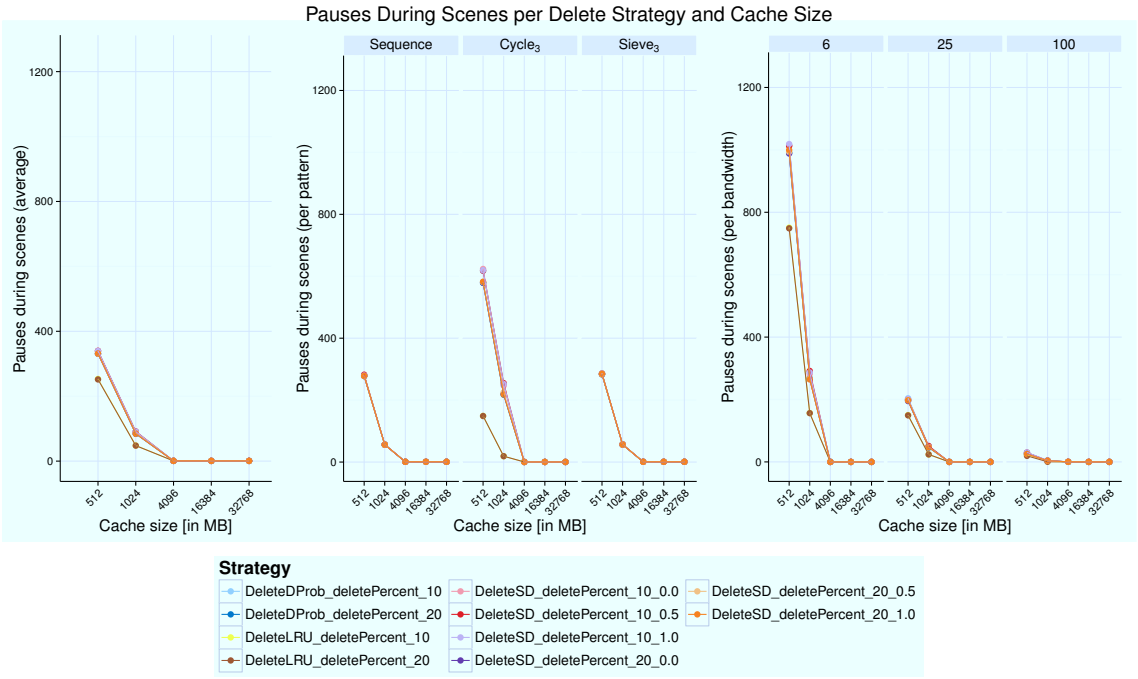


Figure 7.27.: Evaluation of the delete strategies (selected results) - pauses during playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

outliers can be seen in the $cycle_3$ pattern especially when high bandwidths are available (see Appendix I, Figure I.25).

Figure 7.28 shows the average download volume for different bandwidths. A more detailed overview of these results can be found in Appendix I, Figure I.26. The download volume is increasing with increasing bandwidths for all strategies. Thereby, the download volume of the DELETE_LRU strategies is always slightly above the download volume of all other strategies. Grouped by pattern (see Figure 7.28, center), the largest differences between the DELETE_LRU strategies and the other strategies can be recognized for the $cycle_3$ pattern. While the download volume stays about the same for all strategies in the sequence pattern, an increase can be seen for all strategies in the $sieve_3$ pattern. Grouped by cache size (see Figure 7.28, right), it can be noted that the download volume varies only very little from cache size to cache size, but is higher for the DELETE_LRU strategies than it is for the others for small cache sizes.

Figure 7.29 shows the average download volume for different cache sizes. The download volume is decreasing with increasing cache sizes taking a look at small cache sizes. Larger cache sizes reveal constant and nearly equal results for all strategies. Grouped by pattern (see Figure 7.29, center), it can be noted that the $cycle_3$ pattern shows strongly decreasing curves for small cache sizes for all strategies with higher values for the DELETE_LRU strategies. The $sieve_3$ pattern shows even slightly increasing values for the smaller cache sizes for all strategies except the DELETE_LRU strategies. Grouped by bandwidth (see Figure 7.29, right) increasing values can be seen for increasing bandwidths. The larger the bandwidth is, the larger is the difference between the DELETE_LRU strategies and the other strategies for small cache sizes. A more detailed overview over these findings is illustrated in Appendix I, Figure I.27.

Critical reflection: The following statements can be made regarding the overall download volume for the delete strategies:

- The download volume is increasing with increasing bandwidths for all delete strategies.
- The download volume is the same for all strategies for cache sizes of 4096 MB or larger (no delete operations).
- The download volume is higher using the DELETE_LRU strategies in the $cycle_3$ pattern than using other strategies.
- The download volume for the DELETE_LRU strategies is higher using small cache sizes than for the other strategies.

7.5.4. Search for an “Optimal Combination” of Algorithms/Strategies for all Patterns

Figure 7.30 shows the box plots[6] of the average values over all bandwidths, cache sizes, and other parameters like the pattern or the used resolutions and scene lengths for the single pre-fetch strategies and start times. Taking a look at the waiting times at the start of scenes and the number of frames to wait at the start of scenes, it can be noted that the PLAY_STARTUP strategies show lower waiting times at the beginning of a scene for all selected start times. Both PLAY_MIN_RELOAD and PLAY_SCENE strategies require longer waiting times at the beginning of a scene. Thereby, the differences between no pre-fetch and the pre-fetch of at

[6]Please note, that the first and third quartiles are the bottom and the top of the box

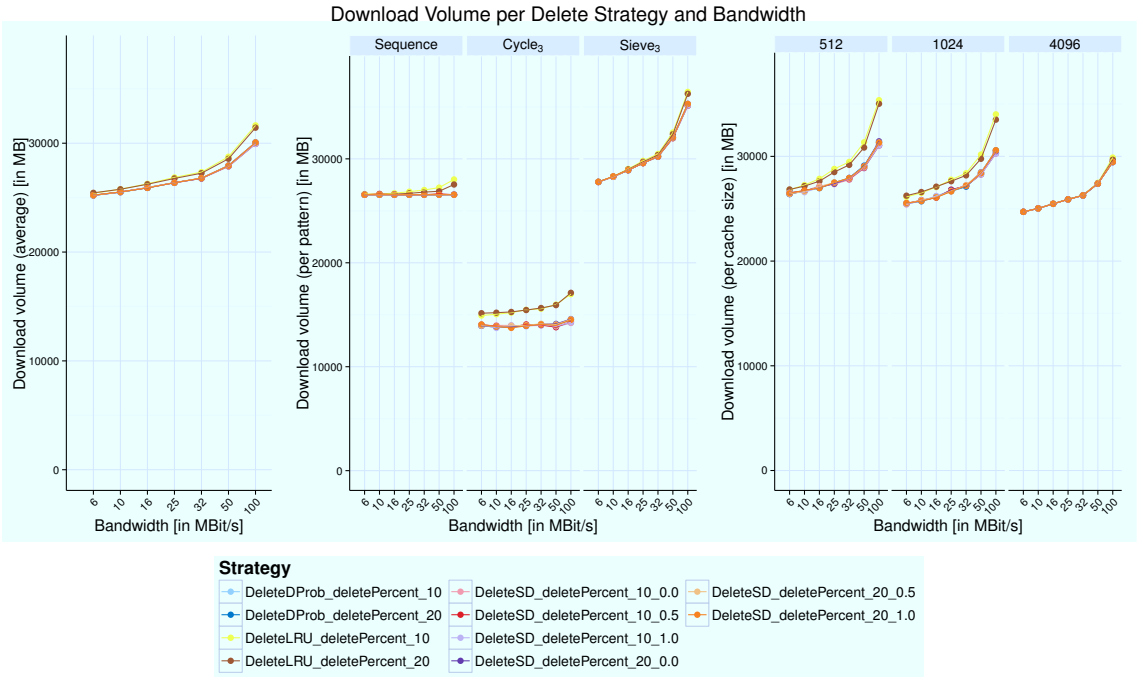


Figure 7.28.: Evaluation of the delete strategies (selected results) - download volume for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

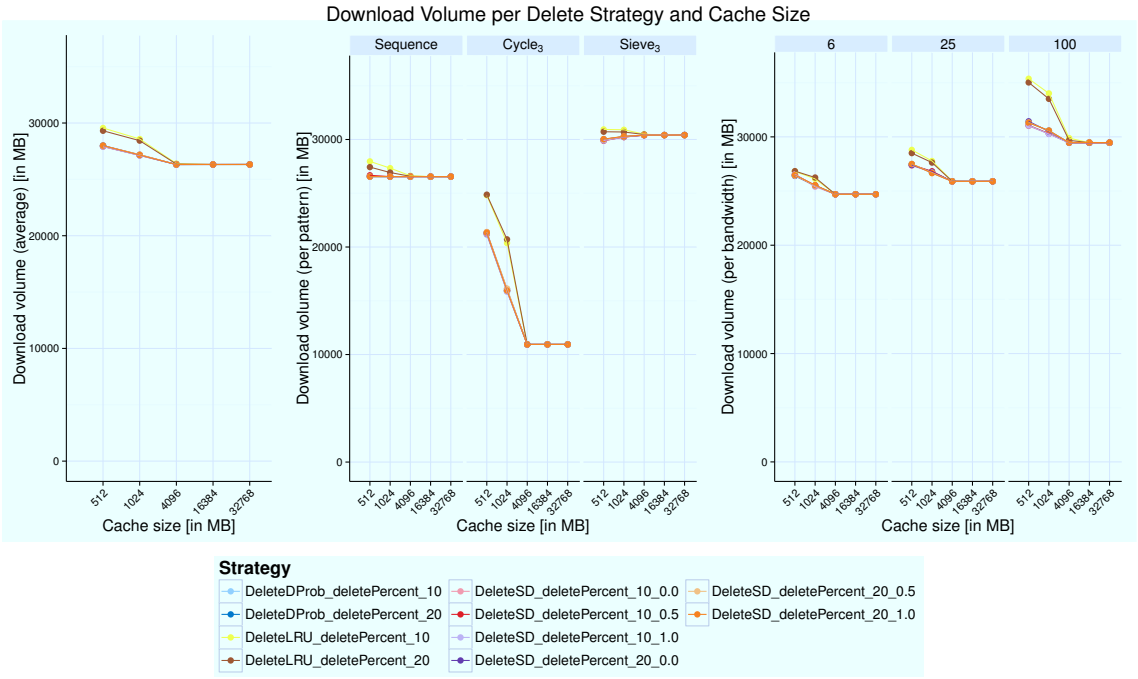


Figure 7.29.: Evaluation of the delete strategies (selected results) - download volume for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

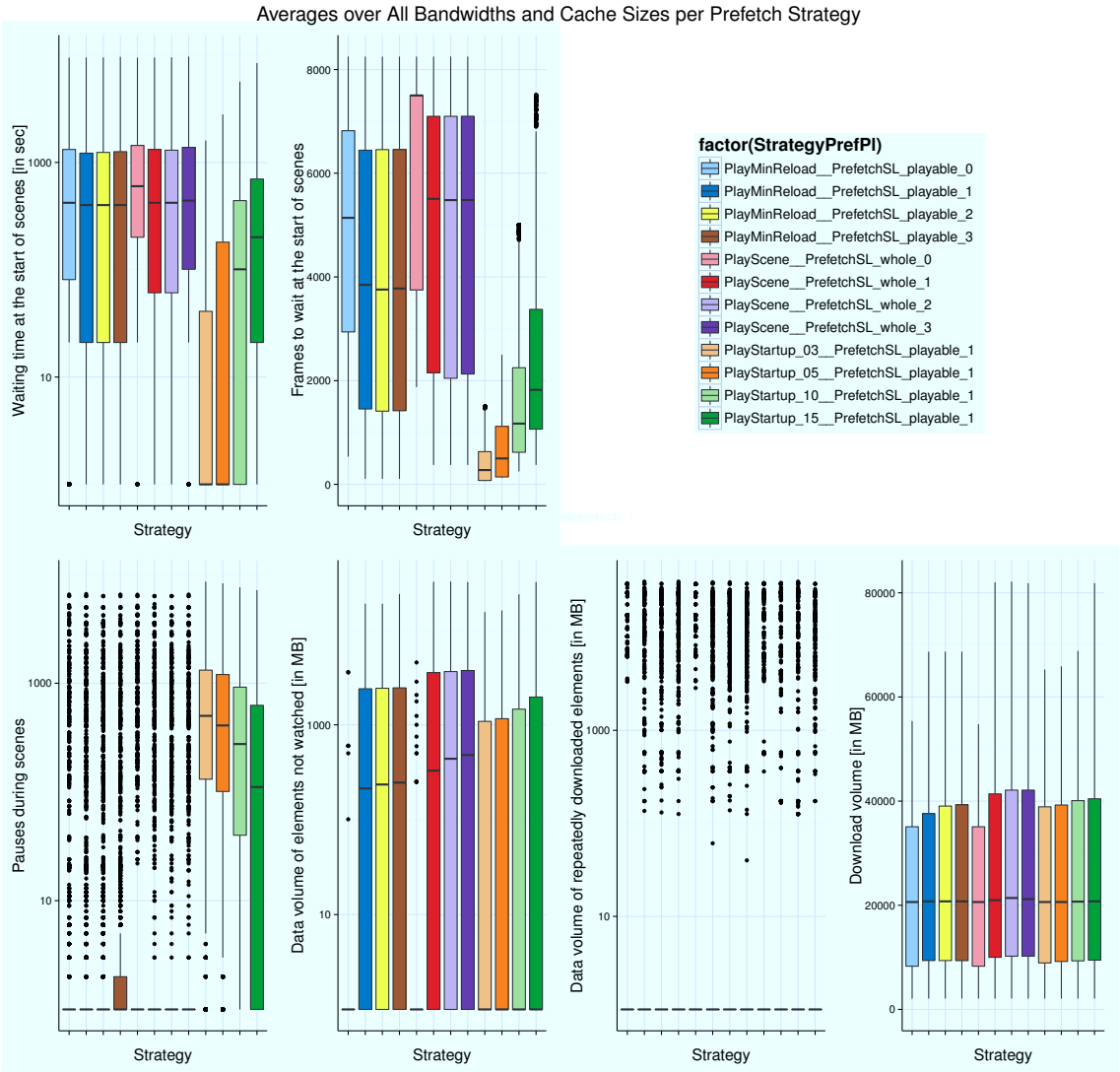


Figure 7.30.: Evaluation of the pre-fetch/start-up strategies - waiting time before playback (upper left) frames to wait before playback (upper center left), pauses during scenes (lower left), data volume of elements not watched (lower center left), data volume of repeatedly downloaded elements (lower center right), and overall data volume (lower right)

least one scene vary significantly while the differences between the pre-fetch of one, two, or three scenes only show very small deviations. While the PLAY_STARTUP strategies revealed quite short waiting times at the beginning of scenes, they require much more pauses during the playback of a scene to download elements for playback. The PLAY_MIN_RELOAD and PLAY_SCENE strategies require a very small but nearly equal average number of pauses and achieve much better results than the PLAY_STARTUP strategies. Taking a look at the data volume of downloaded but elements not watched, it can be noted, that the strategies with no pre-fetch show the best results in this category. The data volume of downloaded but not watched elements increases with increasing start times for the PLAY_STARTUP strategies. Furthermore, the PLAY_MIN_RELOAD strategies achieve a generally smaller number of downloaded but not watched elements than the PLAY_SCENE strategies. Thereby, the strategies with a pre-fetch depth of one scene result in a smaller data volume of elements not watched then the strategies with a pre-fetch of two or three scenes. Regarding the data volume of repeatedly downloaded elements, it can be noted, that the volume increases with increasing start times for the PLAY_STARTUP strategies. It also increases with increasing pre-fetch depths for the PLAY_SCENE strategies. Furthermore, it increases for the PLAY_MIN_RELOAD strategies with pre-fetch. Taking a look at the overall download volume, it can be seen that the strategies with no pre-fetch reveal the lowest download volume (because all elements that are downloaded are displayed to the viewer). The PLAY_STARTUP strategies result in an increasing download volume with increasing start times. The PLAY_MIN_RELOAD strategies with pre-fetch have a smaller download volume than the PLAY_SCENE strategies with pre-fetch.

Figure 7.31 shows the box plots of the results for the delete strategies for the same environmental settings as described for the pre-fetch strategies and start times. The waiting time and the number of frames to wait at the beginning of a scene show that the DELETE_LRU strategies needs longer times than the other strategies to start a scene. All other strategies show only very small or no differences at all. Taking a look at the pauses during a scene, provide the DELETE_LRU strategies better results than the other strategies. Furthermore require the strategies which delete twenty percent of the cache in case the cache is full slightly lesser breaks than the strategies which only delete ten percent of the cache. Regarding the data volume of downloaded but not watched elements, more elements are downloaded for the DELETE_LRU strategies than for all other strategies, which show very similar values. Taking a look at the data volume of repeatedly downloaded elements, the DELETE_LRU strategies have a nearly twice as high download volume than the other strategies, which have nearly identical results. Finally, the download volume of the DELETE_LRU strategies is slightly higher than the download volume of the other strategies.

Depending on the priorities of the viewers, weighted combinations of the to the unit interval standardized values of the number of frames to wait before playback WF_{scene}, the number of pauses during playback P_{sum}, and the overall download volume DLV are built. They may help to decide which combination of strategies should be used in a defined setting or for specified user preferences. Therefor we suggest a combined value η in Equation 7.34 where β can be used to weight the waiting times and the download volume. Depending on the viewers preferences, other functions may be suitable.

$$\eta = \beta \cdot (WF_{scene} + P_{sum}) + (1 - \beta) \cdot DLV,\ \beta \in [0..1] \tag{7.34}$$

Figure 7.32 shows the results for η for the following values of the weighing factor β:

- Combination A: high weight on the waiting times, $\beta = 0.9$

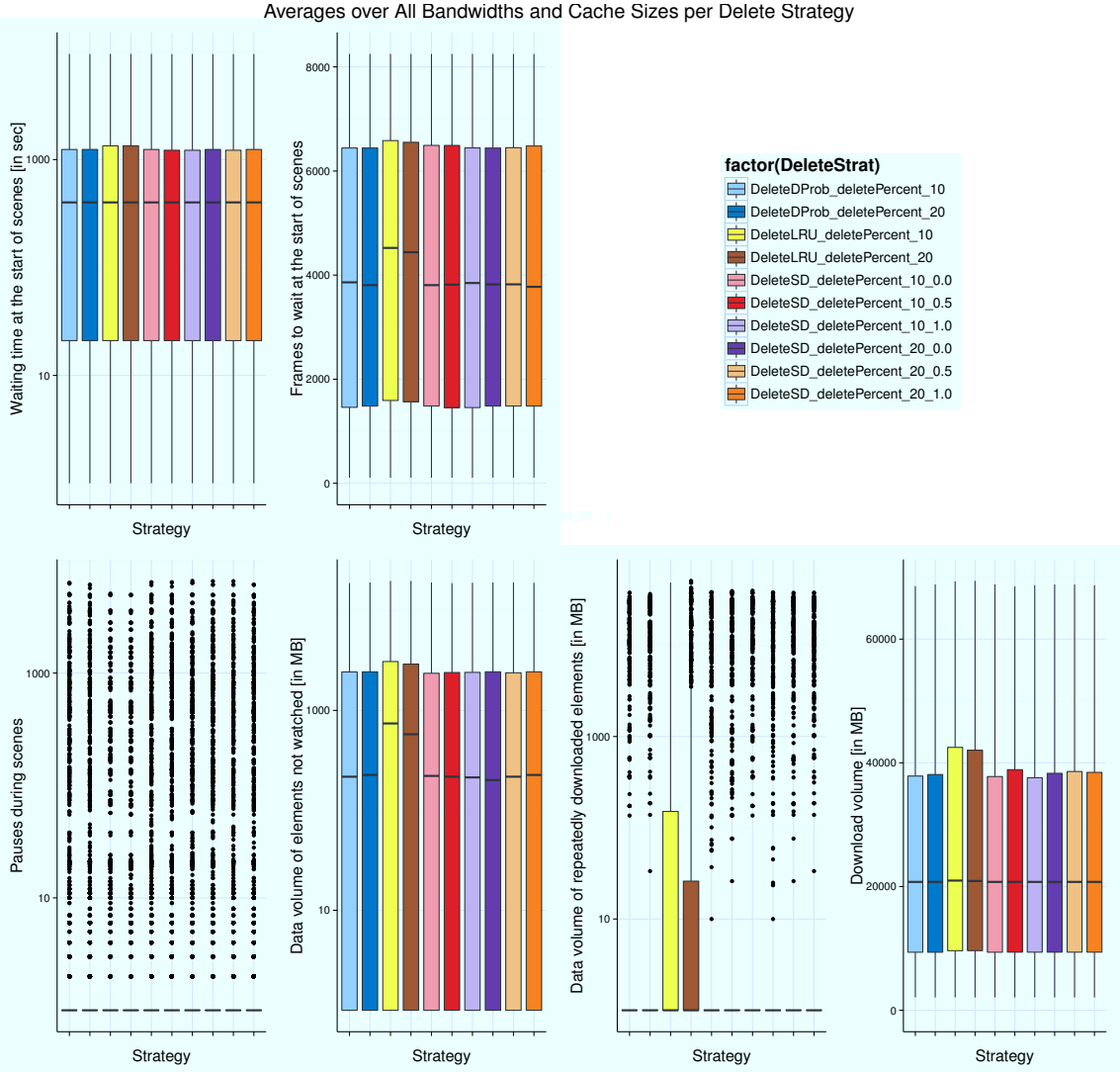


Figure 7.31.: Evaluation of the pre-fetch/start-up strategies - waiting time before playback (upper left) frames to wait before playback (upper center left), pauses during scenes (lower left), data volume of elements not watched (lower center left), data volume of repeatedly downloaded elements (lower center right), and overall data volume (lower right).

- Combination B: equal weights on waiting time and download volume, $\beta = 0.5$
- Combination C: high weight on the download volume, $\beta = 0.1$

Regarding the pre-fetch strategies and start points (see Figure 7.32), it can be noted that depending on the value of β, either the PLAY_MIN_RELOAD strategy with a pre-fetch depth of one scene or the PLAY_MIN_RELOAD strategy with no pre-fetch should be chosen. The PLAY_MIN_RELOAD strategy with a pre-fetch of one scene achieves the best results if the waiting times are highly weighted. The PLAY_MIN_RELOAD strategy with no pre-fetch shows good results if the download volume is weighted at least as high as the waiting times.

Critical reflection: The results of this section can be summarized as follows:

- ***Pre-fetch strategies and start times:***

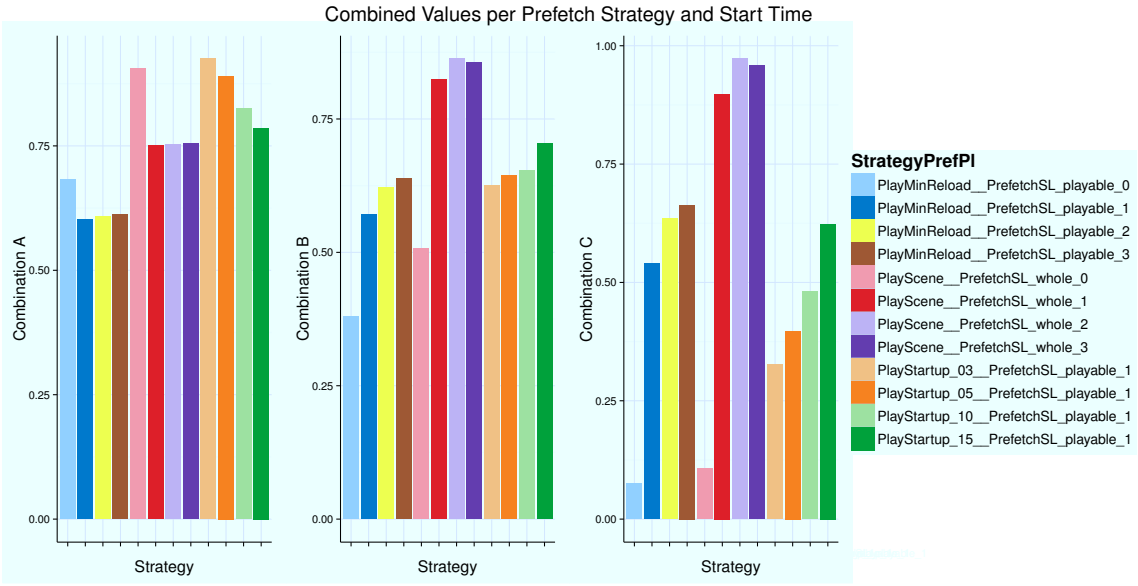


Figure 7.32.: Evaluation of the pre-fetch/start-up strategies combining the number of frames to wait before playback, the number of pauses and the download volume - combination A with a high weight on waiting times (left), combination B with equally weighted values (center), and combination C with a high weight on the overall download volume (right).

  - The pre-fetch strategies and start times with a variable start frame result in higher waiting times at the beginning of scenes than those with a fixed start time.
  - The calculated/determined start frame is smaller for strategies which pre-fetch than for those which do not pre-fetch any elements.
  - The number of pauses is very much higher for strategies with fixed start frames than for those which start playback when the frame is in the cache from which on the scene can be played without pauses.
  - A data volume of zero for the elements not watched can only be achieved if no pre-fetch is used, especially in patterns with forks.
  - The data volume of repeatedly downloaded elements is very small with some outliers with comparatively high values resulting from the $cycle_3$ pattern.
  - The medium of the overall download volume varies only slightly between the different pre-fetch strategies and start times.

- ***Delete strategies:***
  - The delete strategies DELETE_SD and the DELETE_DPROB show very similar values for all metrics.
  - The DELETE_LRU strategy results in slightly higher medium start times than the other strategies.
  - The DELETE_LRU strategy shows slightly higher values for the number of frames to wait at the beginning of scenes, for the data volume of elements not watched

and for the data volume of repeatedly downloaded elements. This results in a slightly higher download volume than for the other strategies.

- ***Weighted strategies:***
  - The PLAY_MIN_RELOAD strategy with a pre-fetch depth of one scene should be used if the viewer wants few waiting times and/or pauses.
  - The best results are achieved for the PLAY_MIN_RELOAD strategy with no pre-fetch if the waiting times and download volume is weighted equally. The same strategy is best if the download volume gets a high weight.

7.5.5. Evaluation of the Strategies for Varying Numbers of Annotations

According to the results from the previous sections, we only use a few algorithms/strategies with significant parameter combinations in this section. While the delete strategies all showed about the same results, we decided to use only the DELETE_SD strategy with $\mu = 1$ and an amount of data to delete of 10 % of the cache size. We furthermore use a pre-fetch depth of one scene, because higher depths did not show any advantages. No tests are performed for the mirrorworld$_3$ and the split$_3$ pattern, because the results for these patterns are very similar to those of the sieve$_3$ pattern. We evaluate the selected strategies from the previous sections with the settings described in the Cartesian product in Equation 7.35 and the sets defined in Equation 7.36. As in the previous sections, we show the results for different bandwidths and cache sizes separately. Hereby, we show the results for each pattern combined with each number of annotations separately to be able to point out certain trends in the data.

$$pattern \times probability \times duration \times anno \times size \times cache \times bw \times pbstart \times pref \times del \tag{7.35}$$

$$\begin{aligned}
pattern &\in \{Cycle_3,\ Sieve_3,\ Sequence\} \subseteq \mathcal{P}atterns \\
probability &= prob_{avg}(p_x) \in \mathcal{P}robabilities \\
duration &\in \left\{dur_{short},\ dur_{medium},\ dur_{long},\ dur_{combi}\right\} \subseteq \mathcal{S}ceneDuration \\
anno &\in \left\{ac_{few},\ ac_{fewmedium},\ ac_{medium},\ ac_{mediummany},\ ac_{many}\right\} \subseteq \mathcal{A}nnoCount \\
size &\in \left\{size_{low},\ size_{medium},\ size_{high}\right\} \subseteq \mathcal{S}izes \\
cache &\in \{512MB,\ 1024MB,\ 4096MB,\ 16384MB,\ 32768MB\} \subseteq \mathcal{C}acheSize \\
bw &\in \{5,76Mbit/s,\ 10Mbit/s,\ 16Mbit/s, \\
&\qquad 25Mbit/s,\ 32Mbit/s,\ 50Mbit/s,\ 100Mbit/s\} \subseteq \mathcal{B}andwidth \\
pbstart &\in \{PLAY_SCENE,\ PLAY_MIN_REL(f_m), \\
&\qquad PLAY_STARTUP(f_x)\} \subseteq \mathcal{P}laybackStart \\
&\qquad f_x \in \{125, 250, 375\} \\
pref &\in \{PREFETCH_SL(z_{SL}, y, \Lambda, dist)\} \subseteq \mathcal{P}refetch \\
&\qquad z_{SL} = 1,\ y \in \{|p_x|, m\},\ \Lambda = 1,\ dist = 1 \\
del &= DELETE_SD(\mu) \in \mathcal{D}elete,\ \mu = 1
\end{aligned} \tag{7.36}$$

Figure 7.33 shows the number of frames to wait at the beginning of scenes for each bandwidth. It can be seen that the numbers of frames to wait are increasing with an increasing number

of annotations for all strategies. Furthermore, the curves are flattening with increasing bandwidths and an increasing number of annotations. Bandwidths of 50 Mbit/s and higher show a significant decrease in the number of frames to wait in the ac_{many} setting especially for the PLAY_MIN_RELOAD and the PLAY_SCENE strategies. This behavior can be explained with the amount of data that can be pre-fetched, which is higher at higher bandwidths and results in smaller numbers of frames to wait accordingly.

The results for the different cache sizes show about the same courses of the curves for each pattern but independent from the number of annotations. Especially for the sequence and the $sieve_3$ pattern, the curves have the same distances to each other. All curves for the $cycle_3$ pattern show the same bend at cache size 4096 MB but the curves are steeper for smaller cache size with higher numbers of annotations. Furthermore, the download volume is increasing with increasing numbers of annotations. The figure illustrating these findings can be found in Appendix I.4, Figure I.28.

The waiting times at the beginning of scenes for each bandwidth show falling curves with growing bandwidths for each pattern and each number of annotations. Thereby only very small differences can be noted between ac_{few}, $ac_{fewmedium}$, and ac_{medium}. Larger differences can be seen between the ac_{medium}, the $ac_{mediummany}$, and the ac_{many} settings for each pattern. The differences between the strategies are high for small bandwidths and many annotations and decrease with increasing bandwidths. These findings are illustrated in Appendix I.4, Figure I.29.

The curves for the different cache sizes are illustrated in Figure 7.34. The curves for the sequence and the $sieve_3$ pattern show about the same behavior where the curves for the PLAY_MIN_RELOAD and the PLAY_SCENE strategies are increasing for the cache sizes 512 MB, 1024 MB, and 4096 MB. They stay at the same value for larger cache sizes. This behavior results from the fact that the scenes fit into the cache using larger cache sizes and the playback for the PLAY_MIN_RELOAD and the PLAY_SCENE strategies can be started at the calculated frame. When the cache is too small for the scene, the playback has to start when the cache is full and pauses will occur as described hereafter. Furthermore are the scenes only played once and some elements have to be downloaded at the beginning of each scene. The curves for the $cycle_3$ pattern does not show this strong increase because all scenes fit into the cache at a certain cache size and the calculated start frames may be in the cache already if the scene is played after the first time. The curves for the PLAY_STARTUP strategies show about the same values for each cache size in each pattern/annotations combination.

As for the waiting time at the beginning of scenes, the bandwidth curves for the pauses during scenes show the same behavior: only very small differences between ac_{few}, $ac_{fewmedium}$, and ac_{medium} and larger differences between the ac_{medium}, the $ac_{mediummany}$, and the ac_{many} settings for each pattern. All curves for the PLAY_STARTUP strategies are falling and lie always above those of the PLAY_MIN_RELOAD and the PLAY_SCENE strategies. The latter ones show significantly better results than the PLAY_STARTUP strategies when the number of annotations increases especially at smaller bandwidths. These findings are illustrated in Appendix I.4, Figure I.30.

Figure 7.35 shows cache size curves for the numbers of pauses during scenes. They are the exact contrast to the cache size curves for the waiting times at the beginning of scenes. These courses verify that the player has to start playback without having all needed elements in the cache, which in turn results in pauses during the scenes. Furthermore, it can be seen that the curves appear in the reverse order of those for the waiting times at the beginning of scenes.

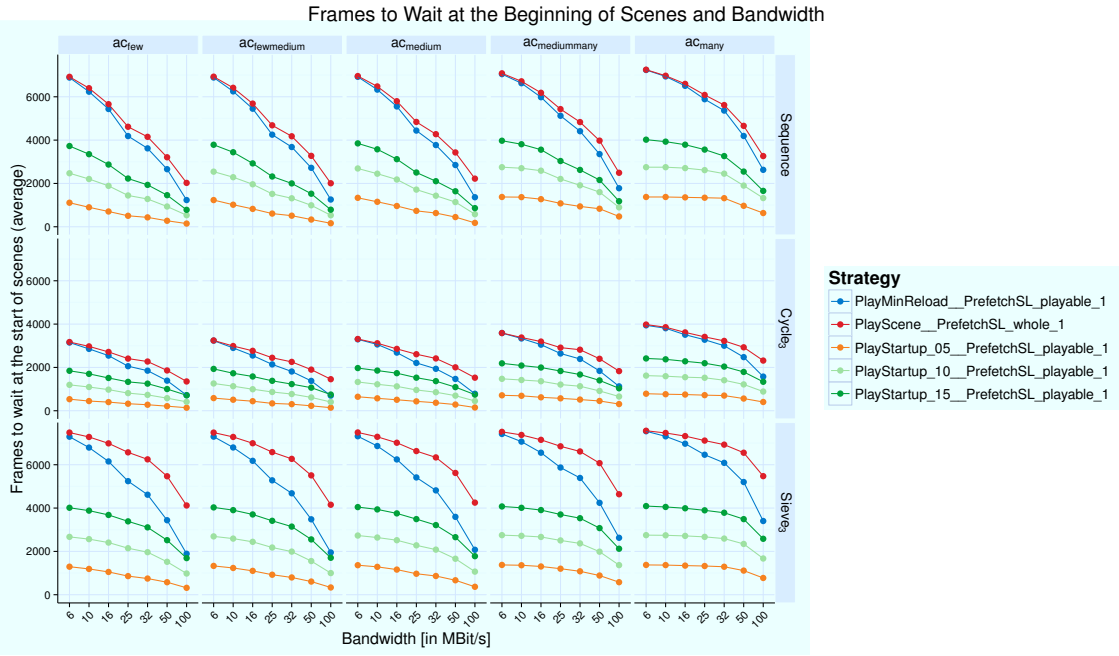


Figure 7.33.: Evaluation of the number of annotations (selected strategies) - frames to wait before playback for different bandwidths: number of annotations × pattern.

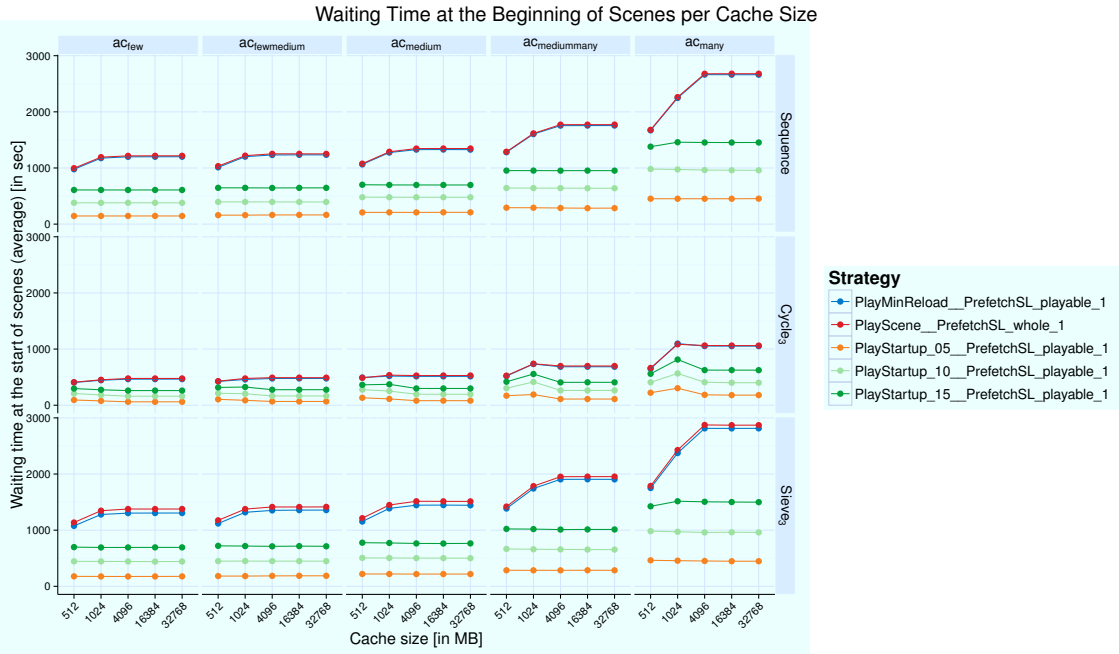


Figure 7.34.: Evaluation of the number of annotations (selected strategies) - waiting time before playback for different cache sizes: number of annotations × pattern.

Taking a look at the data volume of elements not watched, it can be noted that the results vary only slightly between the different tested numbers of annotations within one pattern for all bandwidth curves. A very small number of elements is downloaded but not watched in the sequence and the $cycle_3$ pattern due to the structure of the pattern. Increasing curves can be seen for the $sieve_3$ pattern, but in contrast to the previously contemplated results only very small differences can be noted between the results for the ac_{medium}, the $ac_{mediummany}$, and the ac_{many} settings. Figure I.31 in Appendix I.4 illustrates these findings.

In contrast to the very uniformly looking curves for each pattern for the bandwidths, the cache size curves show greater differences between the different numbers of annotations especially for the $sieve_3$ pattern. Nearly no elements are downloaded but not watched for the scene pattern, but a small amount of data can be recognized for the $cycle_3$ pattern and small cache sizes. The latter behavior can be explained with the pre-fetch of some elements at the single fork in the pattern which have to be deleted before they are watched. The $sieve_3$ pattern shows the same order of the curves for the ac_{few}, the $ac_{fewmedium}$, and the ac_{medium} settings, but the PLAY_MIN_RELOAD and the PLAY_STARTUP strategy with a start time of 15 seconds switch positions for the $ac_{mediummany}$, and the ac_{many} settings. Furthermore, are the curves for the PLAY_MIN_RELOAD and the PLAY_SCENE strategies getting steeper for higher numbers of annotations. This behavior can be explained by the higher amount of data to download in contrast to the small cache sizes.

The curves for the data volume of repeatedly downloaded elements look very similar within one pattern, especially in the sequence and the $sieve_3$ pattern where only very few or no elements at all are downloaded repeatedly. The $cycle_3$ pattern shows variances from bandwidth to bandwidth for all strategies, but no clear tendency. The results fluctuate by a certain amount of data volume but stay at about the same level. An increase of the number of annotations results in an increase in the data volume but the variances stay the same. Taking a look at the different cache sizes, it can be noted that curves for the ac_{few}, the $ac_{fewmedium}$, and the ac_{medium} settings are about the same, but significantly higher values can be seen for the cache sizes 512 MB and 1024 MB for the $ac_{mediummany}$, and the ac_{many} settings. This behavior results from the circumstance that not the whole video can be kept in the cache and elements need to be downloaded repeatedly for each display. Combining both evaluations, it can be concluded that the variations result from the small cache sizes and slightly different points in time where elements are deleted and thus varying amounts of data that are deleted. These findings are illustrated by Appendix I.4, Figure I.32 and Figure I.33.

Taking a look at the bandwidth curves for the overall download volume, it can be seen that the appearance of the curves and the differences between the strategies are very similar within one pattern. Only the download volume for the the $ac_{mediummany}$, and the ac_{many} settings lies significantly above those of the other settings. The download volume of the sequence and the $cycle_3$ pattern shows no or only extremely small variances and reveals constant values within one number of annotations and pattern combination. The download volume for the $sieve_3$ pattern increases with increasing bandwidths which correlates directly with the findings for the downloaded but not watched elements. Thereby, the differences between the strategies are not increasing (see Appendix I.4, Figure I.34).

The cache size curves can be found in Figure 7.37. The curves are on about the same height for the ac_{few}, the $ac_{fewmedium}$, and the ac_{medium} settings. Only their appearance varies from pattern to pattern. Constant values can be seen for the sequence pattern. Increasing and then constant curves are the results from the $cycle_3$ pattern, and slightly increasing curves are

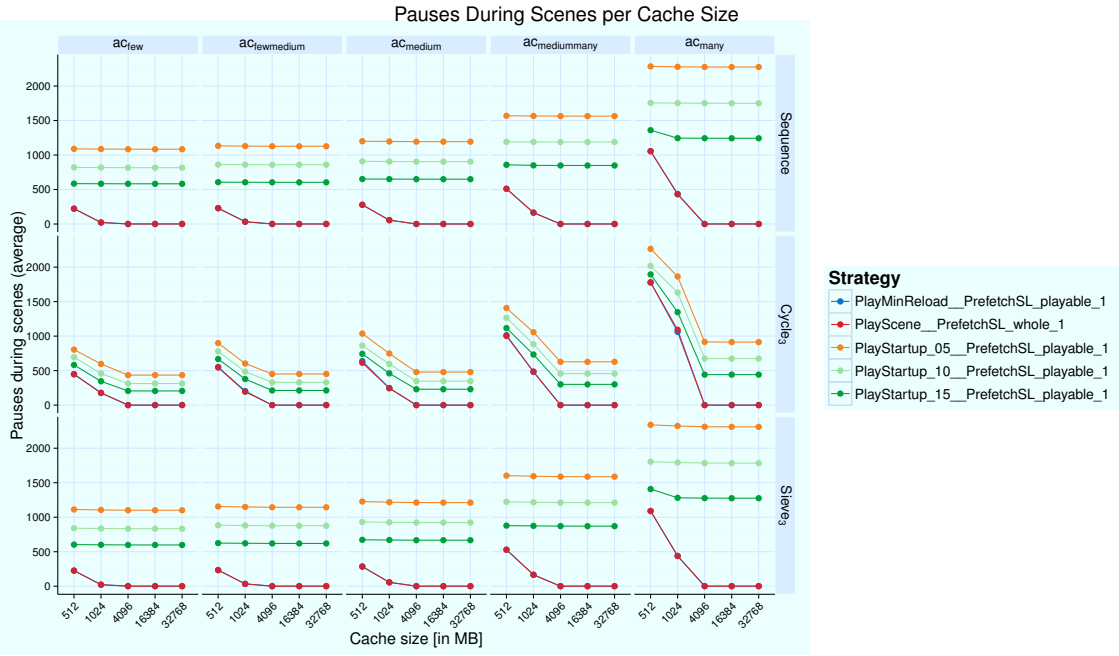


Figure 7.35.: Evaluation of the number of annotations (selected strategies) - pauses during scenes for different cache sizes: number of annotations × pattern.

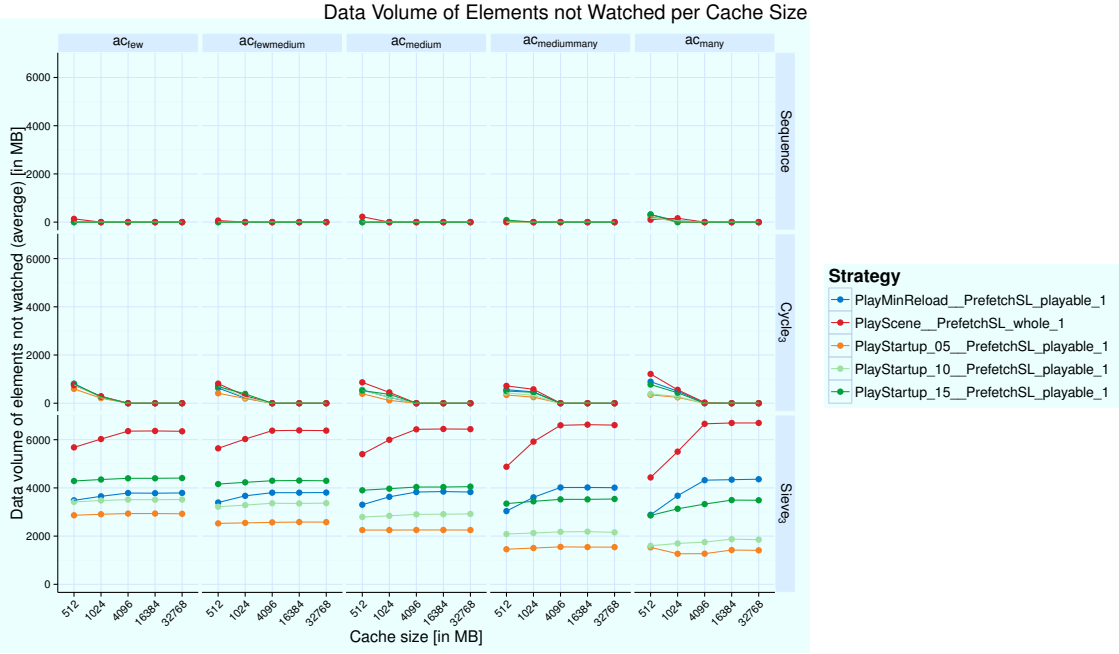


Figure 7.36.: Evaluation of the number of annotations (selected strategies) - data volume of elements not watched for different cache sizes: number of annotations × pattern.

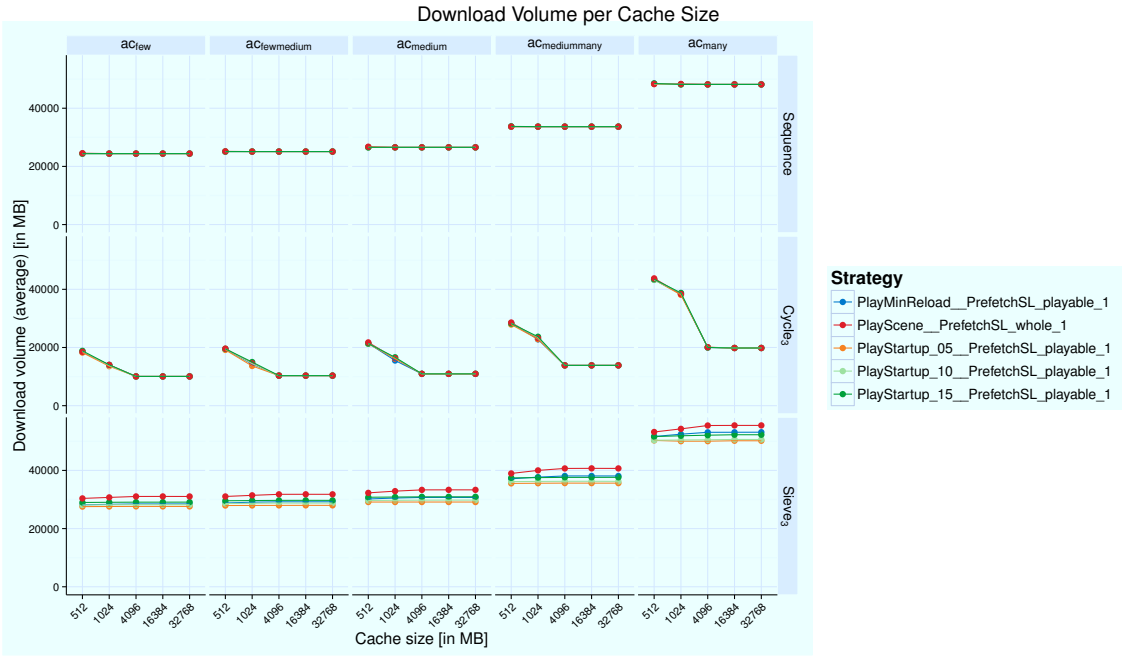


Figure 7.37.: Evaluation of the number of annotations (selected strategies) - download volume the whole video for different cache sizes: number of annotations × pattern.

shown for the sieve$_3$ pattern. Increasing download volumes can be seen for the $ac_{mediummany}$, and the ac_{many} settings without altering the appearance of the curves. The download volume of repeatedly downloaded elements has a strong influence on the overall download volume in the cycle$_3$ pattern, while the download volume of repeatedly downloaded elements is in neither of the considered cases large enough to influence the overall download volume significantly.

Critical reflection: To summarize this section, we can make some generally applicable statements regarding the number of annotations used in one scene. These are as follows:

- The numbers of frames to wait before playback result in falling curves for increasing bandwidths which become less steep with an increasing number of annotations.
- The curves for the waiting times at the beginning of a scene and the pauses during a scene are decreasing with increasing bandwidths.
- The relative position of the curves does not change within one time based metric (WF_{start}, WT_{start}, P_{sum}) for all combinations of numbers of annotations and patterns for all bandwidth curves.
- The curves for the cache sizes are either constant or they are first increasing or decreasing for small cache sizes and then constant for larger cache sizes (where all elements fit into the cache) for the time based metrics.
- More annotations lead to a higher download volume. This results in a change in the position of the curves but does not change their fundamental course for the time based metrics.

- The relative position of the curves for the volume based metrics ($DL_{not\ watched}$, $RDLV$, DLV) only changes between the PLAY_MIN_RELOAD and the PLAY_STARTUP strategy with start time 15 in the $sieve_3$ pattern.
- The appearance of the curves and consequently the behavior of the strategies does not change significantly regarding the volume based metrics, only the degree of increase or decrease may change.

7.5.6. Evaluation of the Strategies for Varying Pattern Widths

In this section, we evaluate the behavior of the selected strategies from the previous section for varying path probabilities and for different widths of the sieve pattern. Used widths are three, five, and seven children of each node. The sieve pattern is chosen because it hast the most forks of all patterns. We evaluate the selected strategies from the previous sections with the settings described in the Cartesian product in Equation 7.37 and the sets defined in Equation 7.38. We discuss the results for bandwidth and cache sizes separately. Hereby, we show the results for each pattern width combined with each used probability separately.

$$pattern \times probability \times duration \times anno \times size \times cache \times bw \times pbstart \times pref \times del \tag{7.37}$$

$$\begin{aligned}
pattern &\in \{Sieve_3,\ Sieve_5,\ Sieve_7\} \subseteq \mathcal{P}atterns \\
probability &\in \{prob_{best}(p_x),\ prob_{bestavg}(p_x),\ prob_{avg}(p_x),\ prob_{worstavg}(p_x), \\
&\qquad prob_{worst}(p_x)\} \subseteq \mathcal{P}robabilities \\
duration &\in \{dur_{short},\ dur_{medium},\ dur_{long},\ dur_{combi}\} \subseteq \mathcal{S}ceneDuration \\
anno &= ac_{medium} \in \mathcal{A}nnoCount \\
size &\in \{size_{low},\ size_{medium},\ size_{high}\} \subseteq \mathcal{S}izes \\
cache &\in \{512MB,\ 1024MB,\ 4096MB,\ 16384MB,\ 32768MB\} \subseteq \mathcal{C}acheSize \\
bw &\in \{5,76Mbit/s,\ 10Mbit/s,\ 16Mbit/s, \\
&\qquad 25Mbit/s,\ 32Mbit/s,\ 50Mbit/s,\ 100Mbit/s\} \subseteq \mathcal{B}andwidth \\
pbstart &\in \{PLAY_SCENE,\ PLAY_MIN_REL(f_m), \\
&\qquad PLAY_STARTUP(f_x)\} \subseteq \mathcal{P}laybackStart \\
&\qquad f_x \in \{125, 250, 375\} \\
pref &\in \{PREFETCH_SL(z_{SL}, y, \Lambda, dist)\} \subseteq \mathcal{P}refetch \\
&\qquad z_{SL} = 1,\ y \in \{|p_x|, m\},\ \Lambda = 1,\ dist = 1 \\
del &= DELETE_SD(\mu) \in \mathcal{D}elete,\ \mu = 1
\end{aligned} \tag{7.38}$$

Figure 7.38 shows the number of frames to wait at the beginning of a scene for each probability and each pattern width for all bandwidths. It can be noted that the results vary in both dimensions, from high to low probabilities and from three to seven children per node, especially for the PLAY_SCENE strategy. The number of frames to wait decreases for increasing bandwidths, but the decrease is the smaller, the smaller the probabilities are. This tendency is the more pronounced the wider the pattern is. Furthermore, the PLAY_MIN_RELOAD strategy is less sensitive to both, pattern width and probabilities regarding the number of frames to wait. Taking a look at the waiting times at the beginning of scenes and the pauses during

scenes for the bandwidths, only very small or no variations at all can be recognized between the different probabilities and the different pattern width (see Appendix I.5, Figure I.36 and Figure I.38).

The cache size curves also show only very small differences between the different probabilities and the different pattern widths. While the numbers of frames to wait and the waiting times at the beginning of scenes are slightly smaller for all strategies for high probabilities, this difference cannot be noticed for the pauses during scenes (see Appendix I.5, Figure I.35, Figure I.37, and Figure I.39).

Significant differences can be seen in the results for the downloaded but not watched elements, both for the analysis of the bandwidth and the cache sizes. Taking a look at the differences between the results of the single strategies for the bandwidth evaluation, it can be noted, that the differences are increasing with decreasing probabilities. At the same time, they are decreasing for increasing pattern widths. The PLAY_SCENE strategy always has the highest values and the PLAY_STARTUP strategy with start time 5 seconds has the lowest values. The other three tested strategies lie in between (see Figure 7.39. Taking a look at the evaluation of downloaded but not watched elements for the cache sizes, the same effect can be seen, the differences are pronounced even more (see Figure 7.40).

The results for the data volume of repeated downloads is not discussed any further here, because extremely few to no repeated downloads occur in this pattern for the used viewer behavior. They neither have an effect on the overall download volume nor can any difference between the different probabilities and the different pattern widths be recognized. The results of the overall download volume are correlated to the results of the download volume of elements not watched, because in this pattern, the overall download volume results from the data volume of downloaded and not watched elements added to the data volume of the watched elements. The latter are the same for each pattern width independent from the probabilities.

Critical reflection: In this section, we analyzed the behavior of the strategies for different pattern width and various probabilities assigned to the paths. The results did not show as many significant differences as the results for the number of annotations but we can nevertheless make some generally applicable statements for the analyzed test cases:

- The number of frames to wait decreases for increasing bandwidths, but the decrease is smaller, the smaller the probabilities are. This tendency is the more pronounced the wider the pattern is.
- The curves for the waiting times at the beginning of a scene and the pauses during a scene show no prominent differences throughout the different probabilities and pattern widths (in the evaluation by cache size and by bandwidth).
- The results for the data volume of repeated downloads are not relevant for the tested pattern.
- The curves for the other volume based metrics ($DL_{not\ watched}$ and DLV) show higher differences with decreasing probabilities (in the evaluation by bandwidth and by cache size).

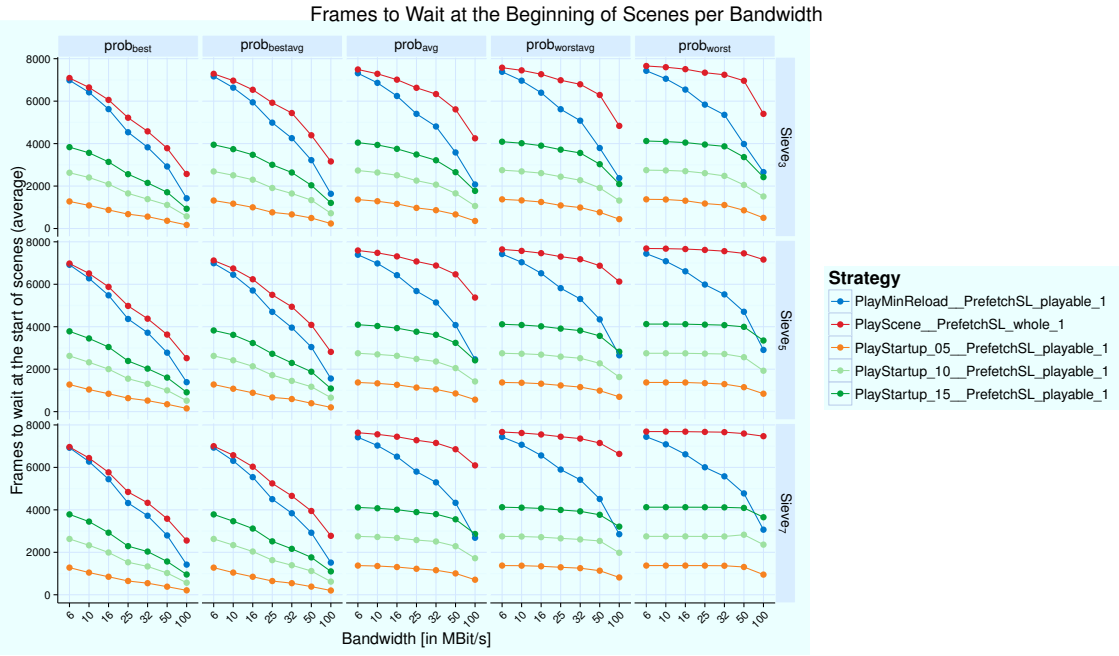


Figure 7.38.: Evaluation of the pattern width and probabilities (selected strategies) - frames to wait before playback for different bandwidths: pattern width × probability.

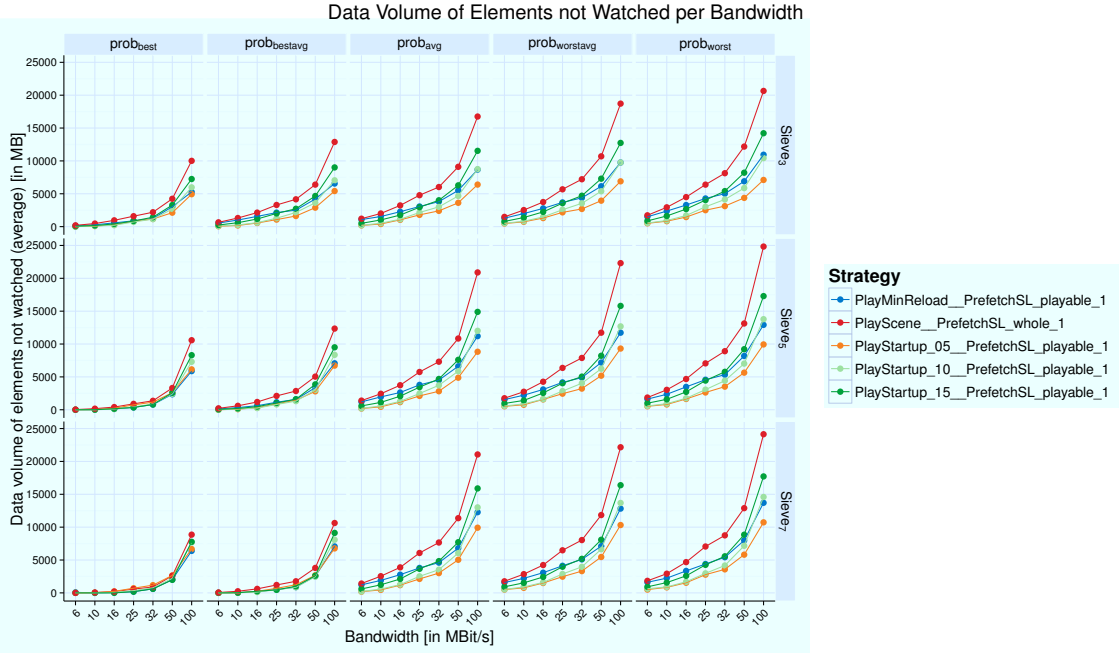


Figure 7.39.: Evaluation of the pattern width and probabilities (selected strategies) - data volume of elements not watched for different bandwidths: pattern width × probability.

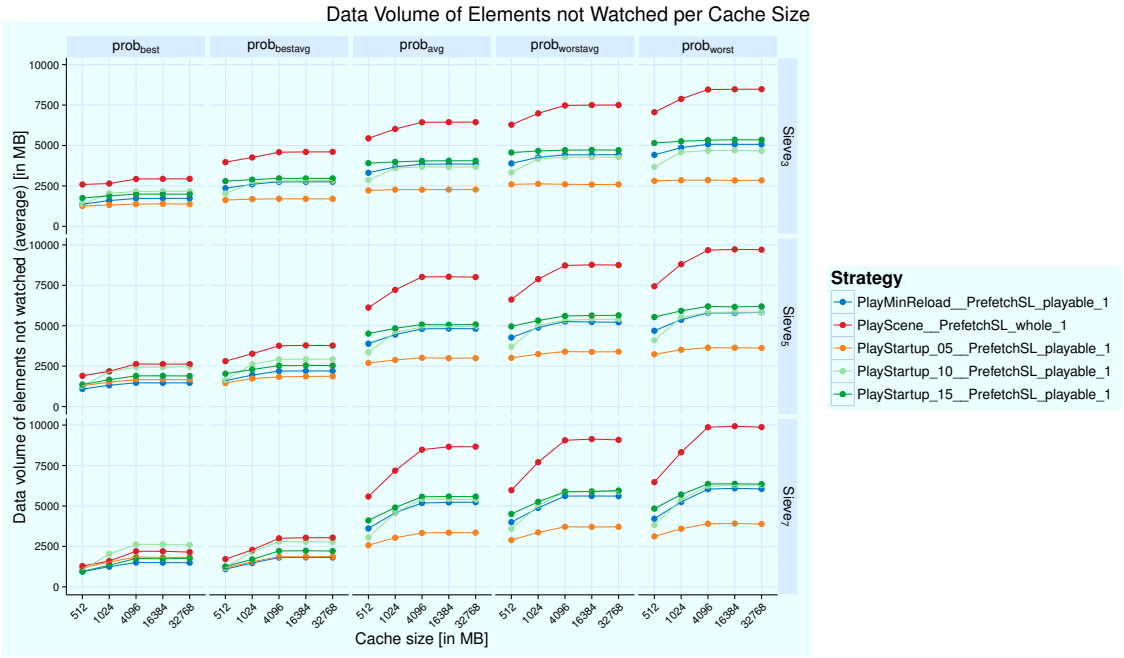


Figure 7.40.: Evaluation of the pattern width and probabilities (selected strategies) - data volume of elements not watched for different cache sizes: pattern width × probability.

7.5.7. Evaluation of the Strategies for the User Generated Scenarios

This section presents the results for the user generated scenarios described in Section 7.2.2 and Appendix F. Therefor, we evaluate the selected strategies from the previous sections with the settings described in the Cartesian product in Equation 7.39 with the sets defined in Equation 7.40.

$$scenario \times pbstart \times pref \times del \tag{7.39}$$

$$\begin{aligned} scenario &\in \{ScenarioA,\ ScenarioB,\ ScenarioC,\ ScenarioD\} \\ pbstart &\in \{PLAY\_SCENE,\ PLAY\_MIN\_REL(f_m), \\ &\quad PLAY\_STARTUP(f_x)\} \subseteq \mathcal{P}laybackStart \\ f_x &\in \{125, 250, 375\} \\ pref &\in \{PREFETCH\_SL(z_{SL}, y, \Lambda, dist)\} \subseteq \mathcal{P}refetch \\ z_{SL} &= 1,\ y \in \{|p_x|, m\},\ \Lambda = 1,\ dist = 1 \\ del &= DELETE\_SD(\mu) \in \mathcal{D}elete,\ \mu = 1 \end{aligned} \tag{7.40}$$

Figure 7.41 shows the numbers of frames to wait for the different tested scenarios. Thereby, Figure 7.41 (left) illustrates all result values as a box plot. Thereby, it can be noted, that the numbers of frames to wait at the beginning of a scene have a greater variation for the PLAY_MIN_RELOAD and the PLAY_SCENE strategies than for the PLAY_STARTUP strategies with different start times. The median of the results is increasing with increasing start times for the PLAY_STARTUP strategies and shows even higher values for the PLAY_MIN_RELOAD and the PLAY_SCENE strategy. This result is consistent with the findings from Section 7.5.4. Taking a look at the individual values, it can be seen that the numbers of frames to wait at the beginning of scenes show significantly higher values for the PLAY_MIN_RELOAD and the PLAY_SCENE strategy in scenarios A, B, and C. Only in scenario D are the differences be-

tween the PLAY_MIN_RELOAD and the PLAY_SCENE strategy less significant compared to the PLAY_STARTUP strategies. Environment three is the only example, where the PLAY_MIN_RELOAD strategy achieves a smaller number of frames to wait than the PLAY_STARTUP strategy with a start time of 15 seconds. The behavior results from small download sizes combined with a high bandwidth.

Figure 7.42 shows the waiting times at the beginning of a scenes for the different tested scenarios. It can be noted that the number of frames to wait does not correlate with the waiting times at the beginning of a scene. This behavior results from the selected bandwidth and cache sizes as well as the chosen resolutions and the color depths. The numbers of frames to wait are for example comparably high in scenario B, while the waiting time is quite small. The difference is especially high in the ((E2) 1680x1050x32 - 100 - 128) setting of scenario B where a comparatively high download volume should be loaded into a small cache. The player has to start playback in the PLAY_MIN_RELOAD and the PLAY_SCENE strategy without having the frame to wait for in the cache. This furthermore results in an increase in pauses. The opposite relationship can seen in scenario C, where larger caches are used with smaller bandwidths. This results in a high number of frames to wait and long waiting times in the last two tested environments. Using an adequate proportion between resolution, color depth, cache size and bandwidth results in simultaneously low numbers of frames to wait and waiting times at the beginning of scenes. The average values of the start times are consistent with the findings in Section 7.5.4.

The number of pauses for all scenarios is illustrated in Figure 7.43. The average numbers of pauses are consistent with the findings in Section 7.5.4. It can be noted, that the number of pauses is high in environments where the number of frames to wait at the beginning of scenes was high, but the waiting time at the beginning of scenes was small (see scenario B (E2) and scenario C (E1)). No or a very small number of pauses occurs for the PLAY_MIN_RELOAD and the PLAY_SCENE strategy, when the scenes fit into the cache (see scenario A (E3) and scenario D (E2, E3)). An equal number of pauses can be observed in scenario C (E3) where a very small cache is used. Only a small amount of the scene fits into the cache while a small bandwidth is used which induces some kind of stop-and-go and results in the equal number of pauses.

The images illustrating the data volume of elements not watched and the data volume of repeatedly downloaded elements can be found in Appendix I.6, Figures I.42 and I.43. The overall download volume for the scenarios is shown in Figure 7.44. These results are consistent with the findings in Section 7.5.4. The data volume of repeatedly downloaded elements as well as the data volume of elements not watched is directly correlated to the download volume except for scenario D (E3) where the values are higher than for (E1) in contrast to the values of the download volume in these environments. It can be noted, that the download volume is the same in the most cases, except in scenario B, where slight variations can be recognized. These variations can be explained with the comparatively high number of scenes and consequently longer paths than in the other scenarios.

Critical reflection: The test of the selected strategies with user generated scenarios revealed the following findings:

- The numbers of frames to wait and the waiting time at the beginning of a scene reflect the findings from the pattern based tests. Depending on the environment settings, the difference between the PLAY_MIN_RELOAD and the PLAY_SCENE strategy in contrast to the PLAY_STARTUP is much higher than seen for the patterns.

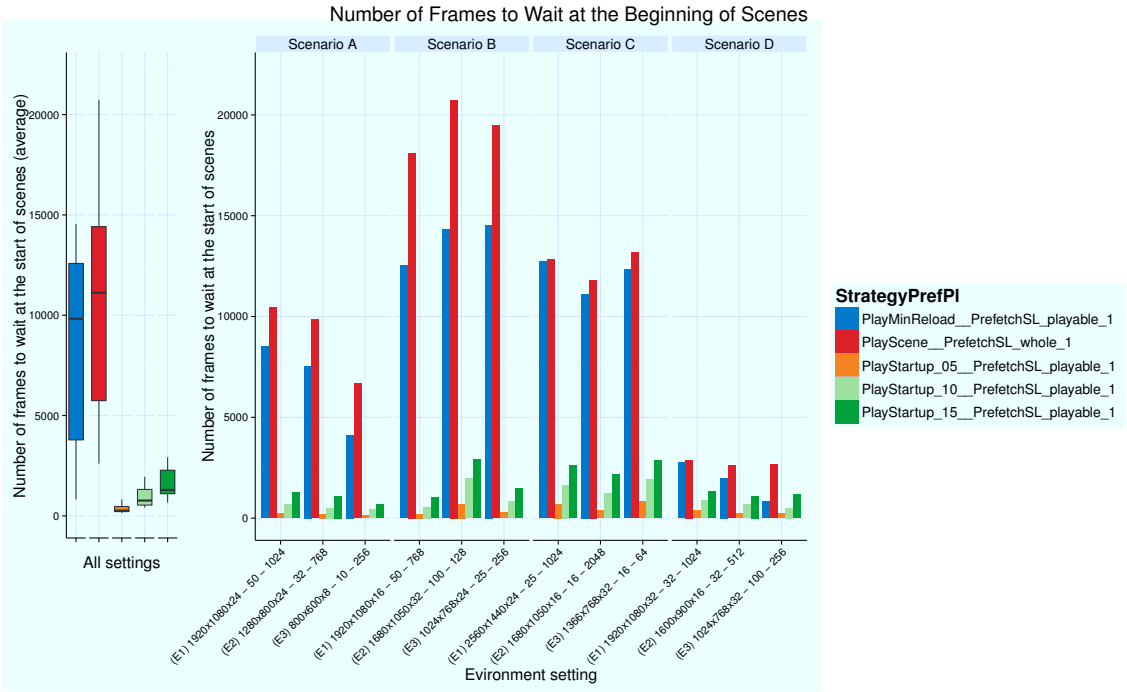


Figure 7.41.: Evaluation of the scenarios - frames to wait before playback: average for the whole test (left) and results grouped by scenario (right).

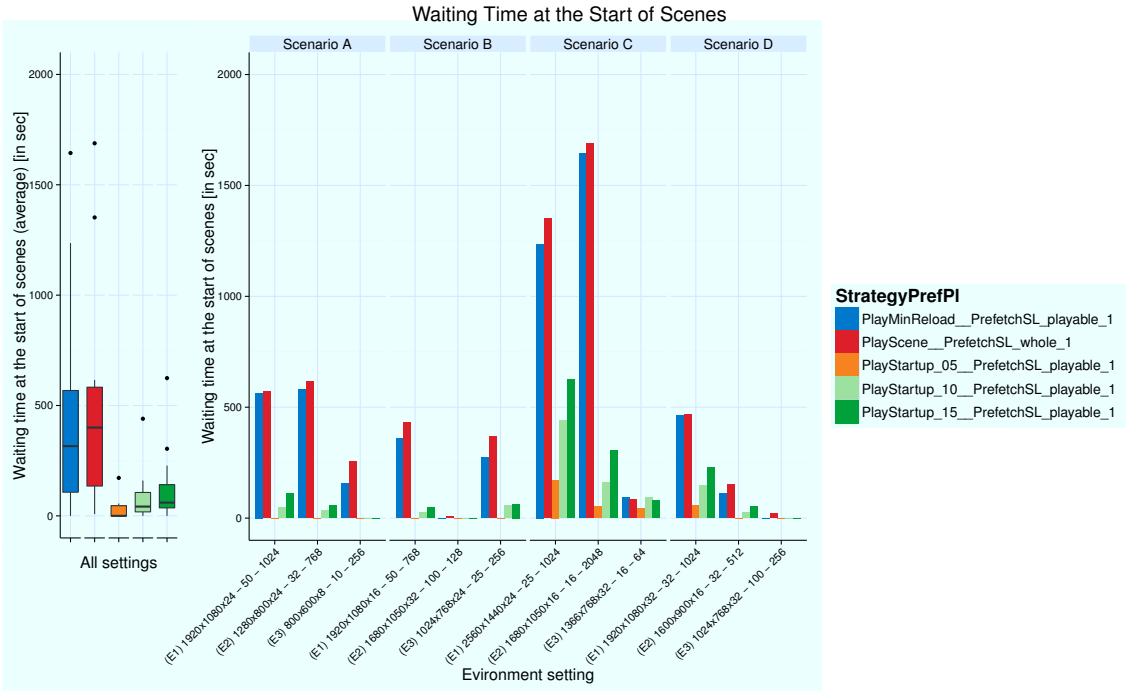


Figure 7.42.: Evaluation of the scenarios - waiting time before playback: average for the whole test (left) and results grouped by scenario (right).

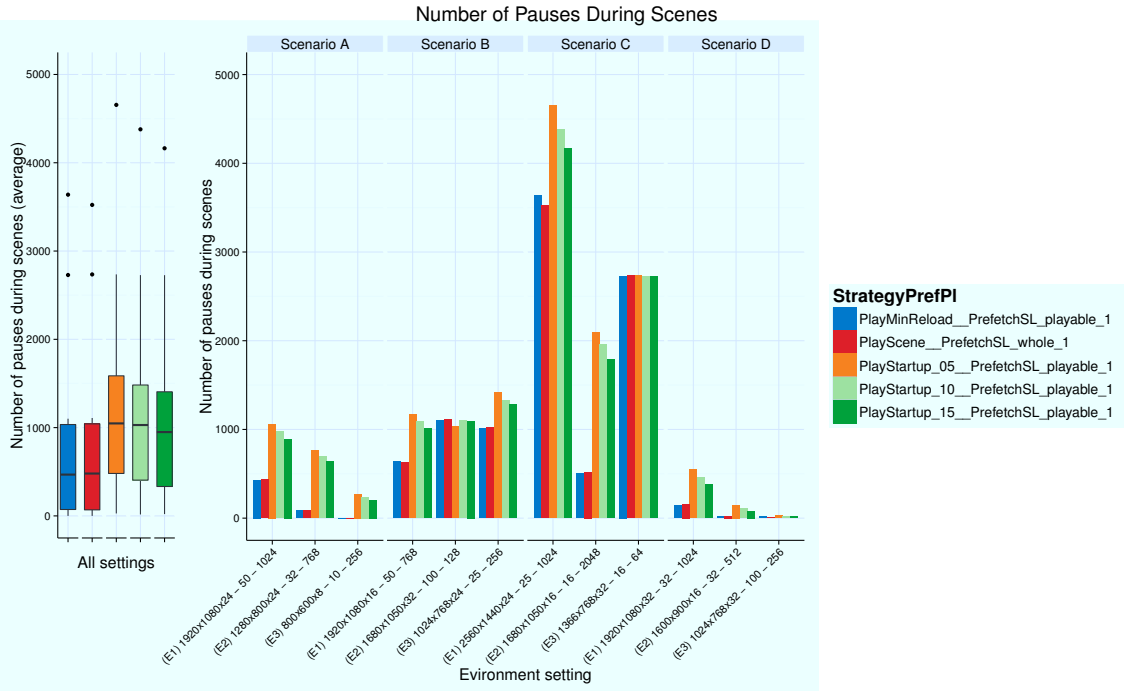


Figure 7.43.: Evaluation of the scenarios - pauses during playback: average for the whole test (left) and results grouped by scenario (right).

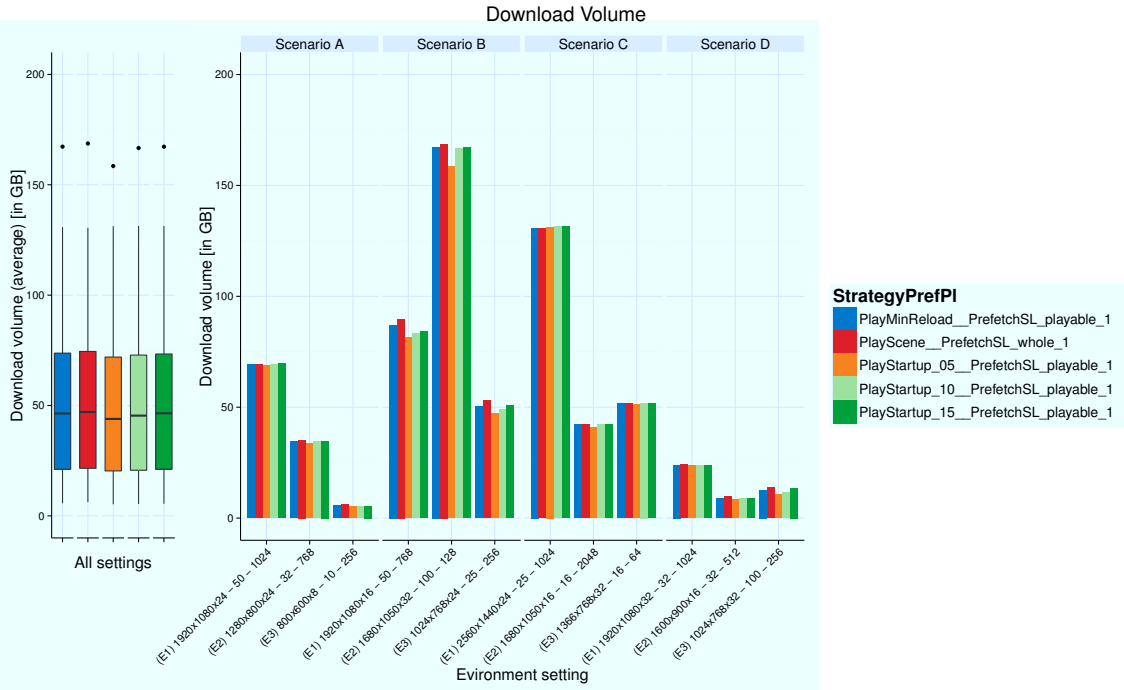


Figure 7.44.: Evaluation of the scenarios - download volume: average for the whole test (left) and results grouped by scenario (right).

- The average number of pauses during scenes also reflects the findings from the tests with the patterns, but the number of pauses may be equal for all strategies in some cases.
- The average download volumes over all patterns are about same for all tested strategies.

7.5.8. Evaluation of the Strategies for Annotations with Varying Priorities

In the previous sections only algorithms/strategies were analyzed which did not take priorities of the elements of an annotated interactive non-linear video into account. This section compares a strategy which is able to deal with different annotation priorities with the strategies evaluated in the previous sections. We therefore evaluate a subset of the strategies with the settings described in the Cartesian product in Equation 7.41 with the subsets defined in Equation 7.42. Due to the fact that the prefetch and deletes strategies were already evaluated in Sections 7.5.2 and 7.5.3 we compare the priority-based strategy only with two other strategies to keep the figures clear. We discuss the PlayMinReload__PrefetchSL_playable_1 and the PlayMinReloadPrio_2__PrefetchSL_playableprio_1_2 strategy in detail, while the PlayStartup_10__PrefetchSL_playable_1 strategy is added to keep the estimation of the differences between the strategies in the right scale. The latter strategy will not be discussed in detail therefore. We analyze the results for various bandwidths and cache sizes. In contrast to the other strategies, we in addition evaluate the $A_{skipped}$ metric which shows how many frames were shown without displaying all annotations.

$$pattern \times probability \times duration \times anno \times size \times cache \times bw \times pbstart \times pref \times del \tag{7.41}$$

$$\begin{aligned}
pattern &\in \{Cycle,\ Mirrorworld,\ Sieve,\ Split,\ Sequence\} \subseteq \mathcal{P}atterns \\
probability &\in \left\{prob_{best}(p_x),\ prob_{avg}(p_x),\ prob_{worst}(p_x)\right\} \subseteq \mathcal{P}robabilities \\
duration &\in \left\{dur_{short},\ dur_{medium},\ dur_{long},\ dur_{combi}\right\} \subseteq \mathcal{S}ceneDuration \\
anno &\in \left\{ac_{few},\ ac_{fewmedium},\ ac_{medium},\ ac_{mediummany},\ ac_{many},\ ac_{combi}\right\} \subseteq \mathcal{A}nnoCount \\
size &\in \left\{size_{low},\ size_{medium},\ size_{high}\right\} \subseteq \mathcal{S}izes \\
cache &\in \{512MB,\ 1024MB,\ 4096MB,\ 16384MB,\ 32768MB\} \subseteq \mathcal{C}acheSize \\
bw &\in \{5,76Mbit/s,\ 10Mbit/s,\ 16Mbit/s, \\
&\qquad 25Mbit/s,\ 32Mbit/s,\ 50Mbit/s,\ 100Mbit/s\} \subseteq \mathcal{B}andwidth \\
pbstart &\in \{PLAY_SCENE,\ PLAY_MIN_REL(f_m), \\
&\qquad PLAY_MIN_REL_PRIO(f_{m/n}, \Lambda),\ PLAY_STARTUP(f_x)\} \subseteq \mathcal{P}laybackStart \\
&\qquad \Lambda \in \{1,2\},\ f_x \in \{250\} \\
pref &\in \{PREFETCH_SL(z_{SL}, y, \Lambda, dist)\} \subseteq \mathcal{P}refetch \\
&\qquad z_{SL} = 1,\ y \in \{|p_x|, m, n\},\ \Lambda \in \{1,2\},\ dist = 1 \\
del &\in \{DELETE_SD(\mu),\ DELETE_PRIO(\Lambda)\} \subseteq \mathcal{D}elete \\
&\qquad \mu = 1,\ \Lambda \in \{1,2\}
\end{aligned} \tag{7.42}$$

As in the other evaluations, we first take a look at the number of frames to wait before the playback starts (WF_{start} metric). Figure 7.45 shows the average number of frames to wait

as well as the results for different patterns and path probabilities at the start of scenes for different bandwidths. A more detailed overview can be found in Appendix I.7, Figure I.44. It can be seen that the number of frames to wait for the priority-based strategy is smaller than for the PLAY_MIN_REL strategy with download depths of one scene. This behavior results from the fact that not all annotations have to be considered for the calculation of the start frame which results in a smaller index number of the frame. Grouping the data for the different patterns, it can be noted that the start frame of the priority-based strategy differs the most from the PLAY_MIN_REL strategy in the cycle pattern at small bandwidths. Comparing the strategies with regard to different path probabilities, no big differences between a high and a low probabilities can be noted.

Figure 7.46 shows the number of frames to wait at the beginning of scenes for different bandwidths. A more details overview can be found in Appendix I.7, Figure I.45. All strategies result in a downward tendency for cache sizes smaller than 4096 MB. Thereby the differences between the PLAY_MIN_REL strategy and the priority-based strategy are about the same for the sequence and the sieve_3 pattern. In contrast to that are the differences smaller for cache sizes higher than 4096 MB smaller. Taking a look at the results for the grouping over the bandwidths, it can be seen that the differences between the PLAY_MIN_REL strategy and the priority-based strategy are very small for small bandwidths and the larger, the higher the bandwidth is. As described in Section 7.5.2, this behavior results from the algorithm for calculating the start frame. The smaller the bandwidth is, the more elements need to be downloaded before starting the playback tho achieve a playback without pauses.

While the number of frames to wait is a calculated value, the WT_{start} metric shows the time the user has to wait taking the filling of the cache into account. Figure 7.47 shows the results for the waiting times at the beginning of scenes (for a more detailed overview see Appendix I.7, Figure I.46). It can be seen that the average waiting time for the priority-based strategy is always below that of the PLAY_MIN_REL strategy and a prefetch depth of one scene. The difference between these two values get the smaller, the higher the bandwidth is. This behavior can be noticed for all tested patterns. Taking a look at the number of annotations in a scene, it can be seen that the difference between the two strategies gets the larger the more annotations are in a scene (and thus may be dropped during the calculation of the start frame, which results in a smaller start time). The evaluation of the values for the different cache sizes does not show any significant results. It can be stated that the waiting times differ more when the values are grouped by bandwidth for small bandwidths. The difference decreases with increasing bandwidths. The illustration of these results can be found in Appendix I.7, Figure I.47.

Taking a look at the number of pauses during scenes, it can be stated that for the priority-based strategy, the value for the P_{sum} metric is smaller or equal to the PLAY_MIN_REL strategy and a prefetch depth of one scene. These results are illustrated in Figure 7.48 and more precisely in Appendix I.7, Figure I.48 and Figure I.49. Thereby this difference is larger in the cycle pattern than in the other patterns for the different bandwidths and cache sizes. The difference furthermore decreases with increasing cache sizes and bandwidths.

The number of frames with skipped annotations is expressed by the $A_{skipped}$ metric. The results for this metric are illustrated in Figures 7.49 and 7.50, and more detailed in Appendix I.7, Figure I.50 and Figure I.51. The results for the PLAY_MIN_REL strategy and a prefetch depth of one scene are zero skipped frames for all cache sizes and all bandwidths in every tested pattern setting. In this strategy (and the other strategies evaluated so far), it is not intended

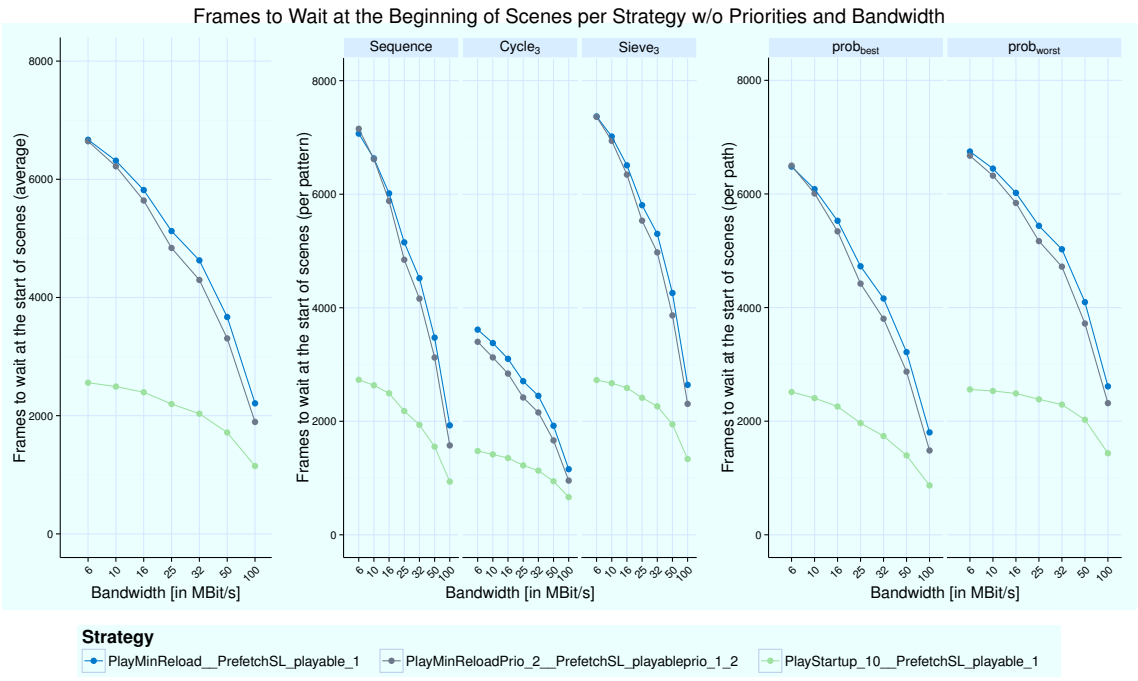


Figure 7.45.: Evaluation of the priority-based strategies (selected results) - frames to wait before playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by used probabilities (right).

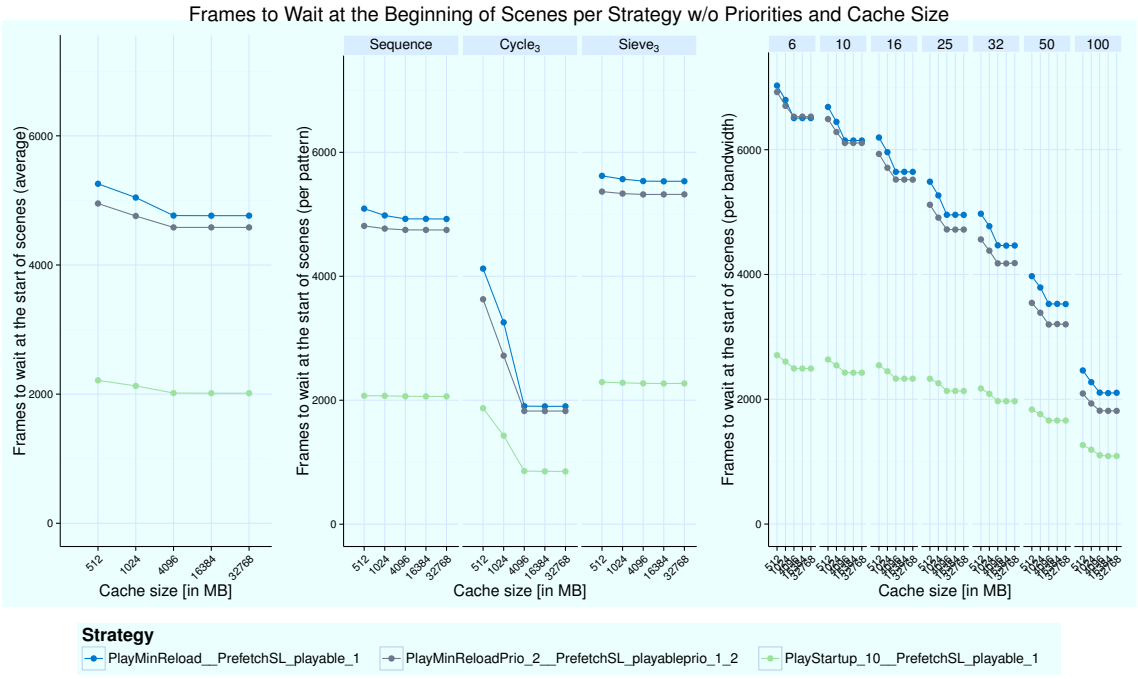


Figure 7.46.: Evaluation of the priority-based strategies (selected results) - frames to wait before playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by used probabilities (right).

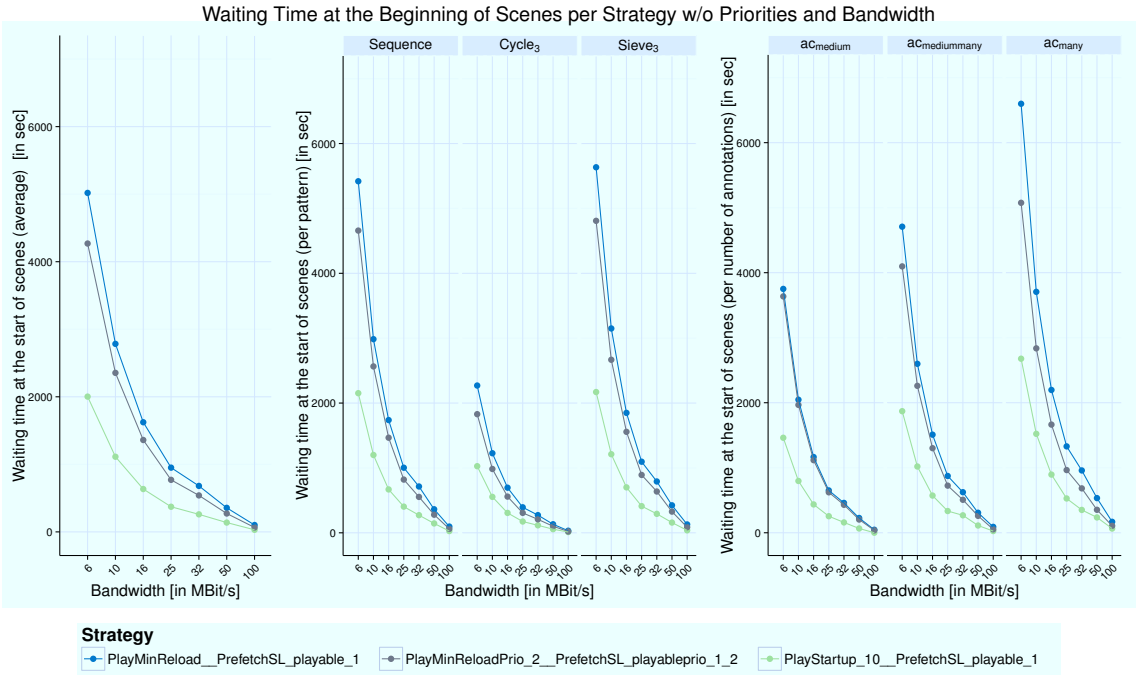


Figure 7.47.: Evaluation of the priority-based strategies (selected results) - waiting time before playback for different bandwidths: average for the whole test (left) and results grouped by pattern (right).

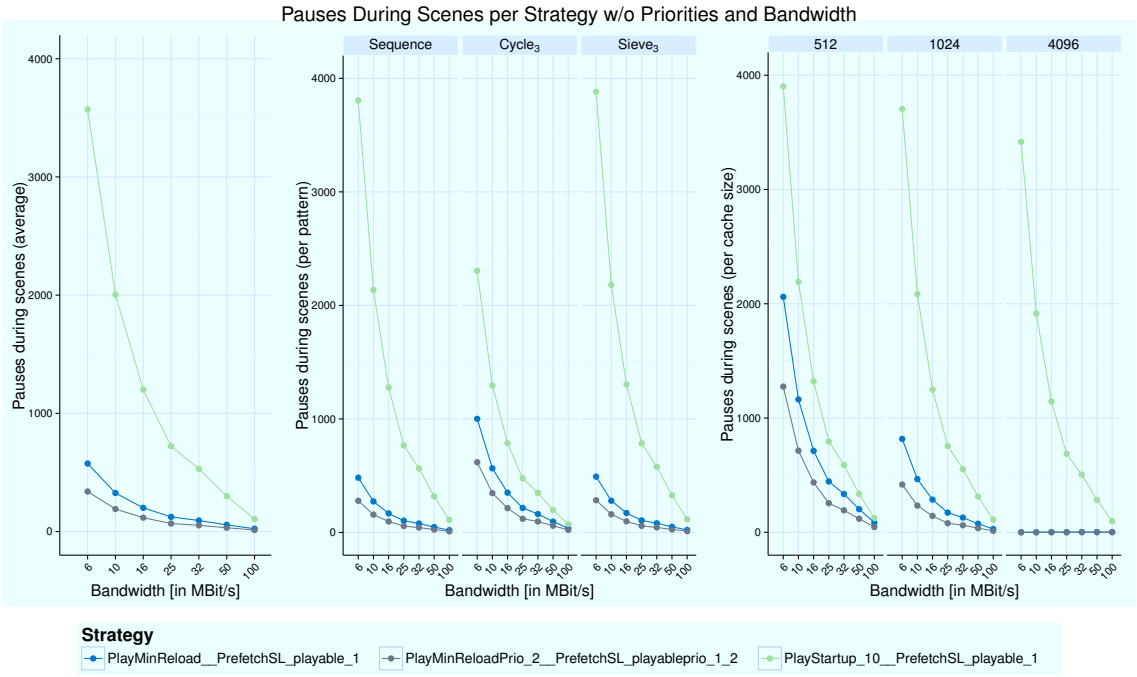


Figure 7.48.: Evaluation of the patterns (selected results, priority-based strategy and start time) - pauses during playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

to omit annotations from being displayed. All annotations are downloaded and displayed to the viewer. For that reason, we only discuss the results for the priority-based strategy hereafter. It can be stated, that the number of frames with skipped annotations stays about the same for bandwidths below 32 Mbit/s. The number is only decreasing for higher bandwidths. Thereby, the difference between the cycle and the other patterns is not as high as it can be noticed for other metrics. The cache sizes have influence on this metric in the following way: when the cache sizes are to small to store all needed elements, the number of frames with skipped annotations is higher then for cache sizes which fit all elements. This effect is more pronounced for the cycle_3 pattern than for the other patterns.

Regarding the data volume of downloaded but not watched elements ($DL_{not\ watched}$), it can be stated that the results for the priority-based strategy are similar to the PLAY_MIN_REL strategy and a prefetch depth of one scene. In contrast to the metrics evaluated so far in this subsection, the results of the priority-based strategy are not always better or equal to those of the PLAY_MIN_REL strategy and a prefetch depth of one scene. As shown in Figures 7.51 and 7.52 (and more detailed in Appendix I.7, Figure I.52 and Figure I.53), it can be seen that the priority-based strategy achieves higher values in the sequence pattern grouping the results by pattern, and for high path probabilities grouping the results by path probabilities for both, small bandwidths and small cache sizes. The differences between the results of both strategies are increasing with increasing cache sizes for the sieve_3 pattern.

The results for the data volume of repeatedly downloaded elements ($RDLV$) can be seen in Figures 7.53 and 7.54, as well as in Appendix I.7, Figure I.54 and Figure I.55. It can be noted that data volume of repeatedly downloaded elements is smaller or equal for the priority-based strategy than for the strategy with a PLAY_MIN_REL strategy and a prefetch depth of one scene for every bandwidth and every cache size regardless the considered settings for the patterns. Thereby, the differences are very small for the sequence and the sieve_3 pattern and high for the cycle_3 pattern grouping both, results for the cache sizes and the bandwidths, by pattern. The results for the priority-based strategy are significantly better in the cycle_3 pattern compared to the strategy with a PLAY_MIN_REL strategy and a prefetch depth of one scene. The difference results from the values achieved for the small cache sizes of 512 MB and 1024 MB. Thereby, it increases with increasing bandwidths.

Taking a look at the overall download volume (DLV), it can be stated, that the results for the priority-based strategy are better than for the strategy with a PLAY_MIN_REL strategy and a prefetch depth of one scene for each evaluated bandwidth and cache size independent from the used pattern settings. The results are illustrated in Figures 7.55 and 7.56 and in more detail in Appendix I.7, Figure I.56 and Figure I.57. Taking a look at the results for the different bandwidths, it can be seen that the difference between both strategies is about the same independent from the used pattern. Taking a look at the cache sizes, the differences vary for the cycle_3 pattern while the differences are about the same for the other patterns. In contrast, the number of annotations has a large influence on the difference between both strategies. It is increasing with an increasing number of annotations independent from the regarded bandwidth and cache size.

Critical reflection: To summarize this section, the following results can be summarized for the strategy using a PLAY_MIN_REL strategy and a pre-fetch depth of one scene compared to the priority-based strategy:

- The frame to wait for is always smaller for the priority-based strategy than for the PLAY_MIN_REL strategy and a pre-fetch depth of one scene.

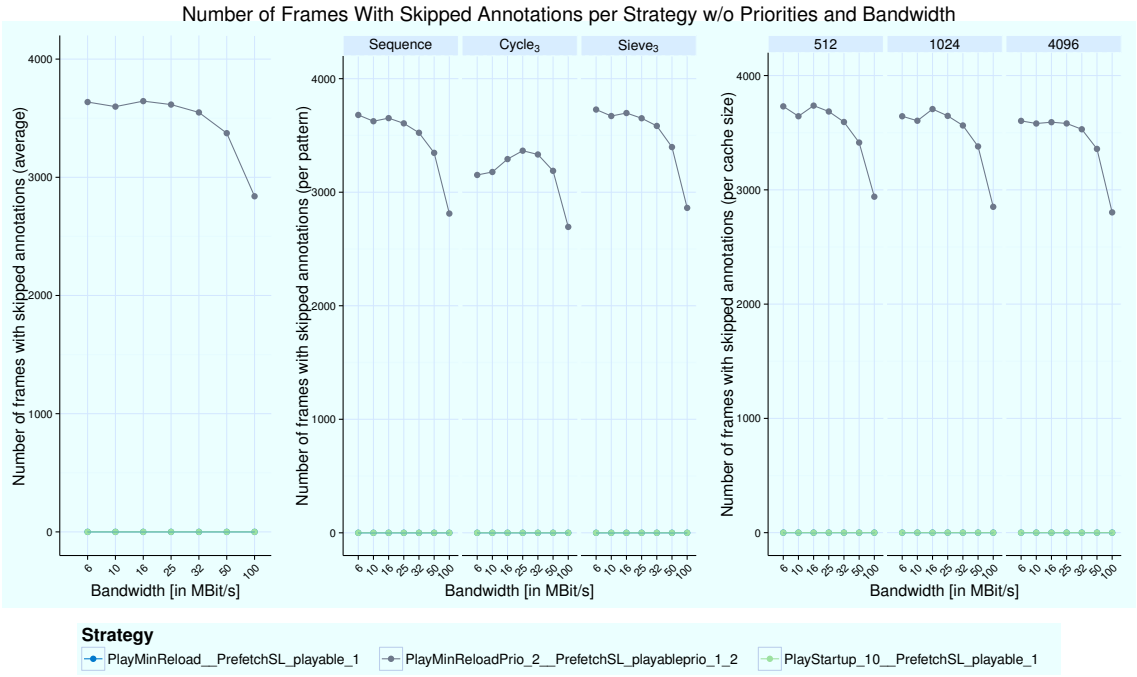


Figure 7.49.: Evaluation of the patterns (selected results, priority-based strategy and start time) - pauses during playback for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

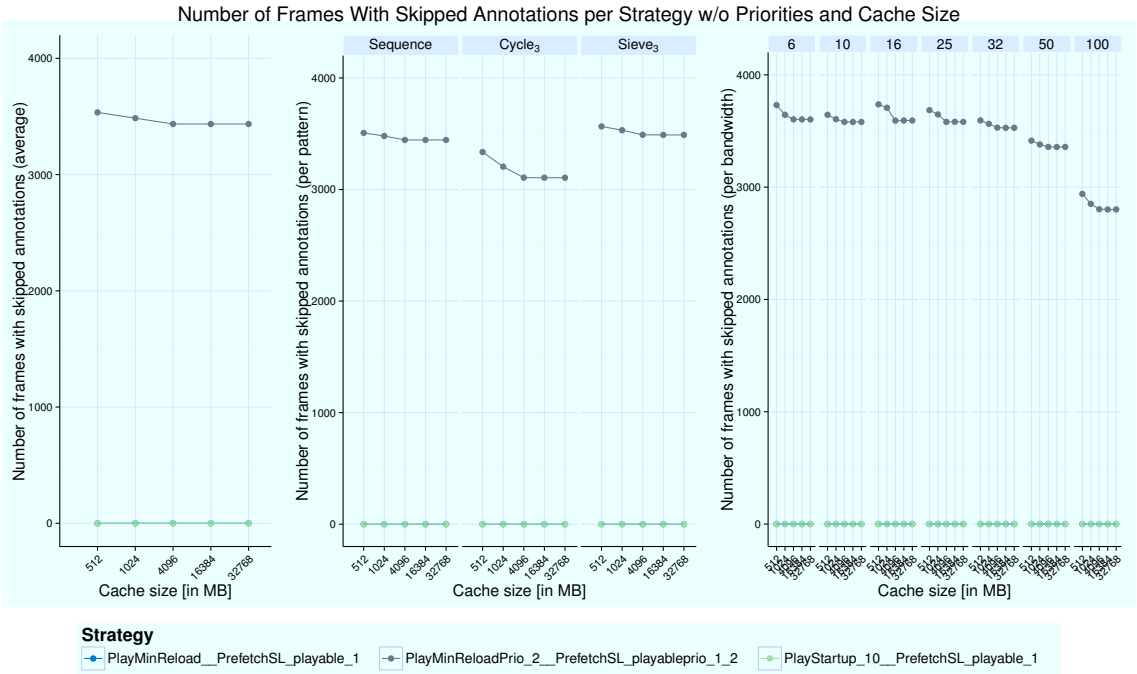


Figure 7.50.: Evaluation of the patterns (selected results, priority-based strategy and start time) - pauses during playback for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

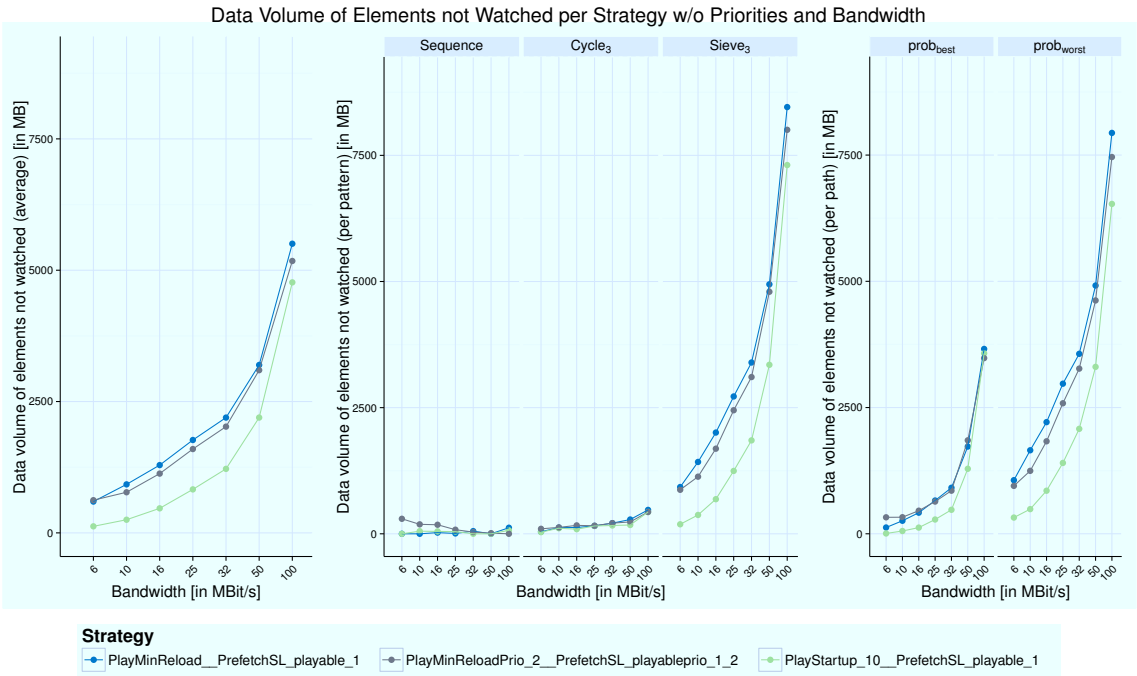


Figure 7.51.: Evaluation of the patterns (selected results, priority-based strategy and start time) - data volume of elements not watched for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

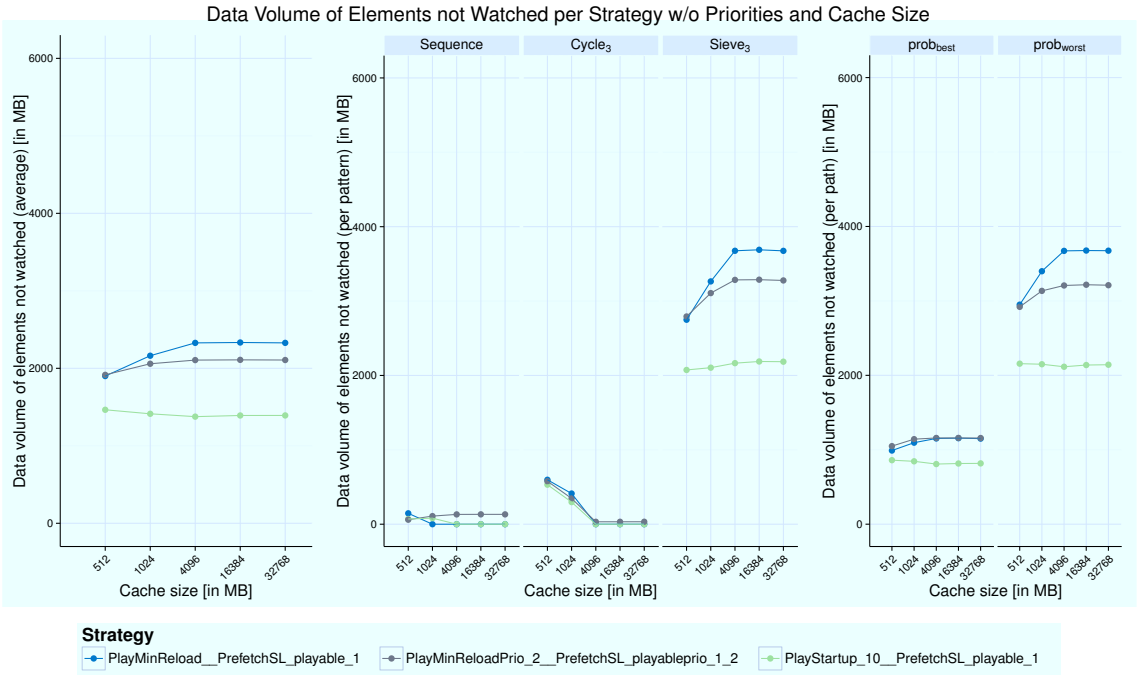


Figure 7.52.: Evaluation of the patterns (selected results, priority-based strategy and start time) - data volume of elements not watched for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

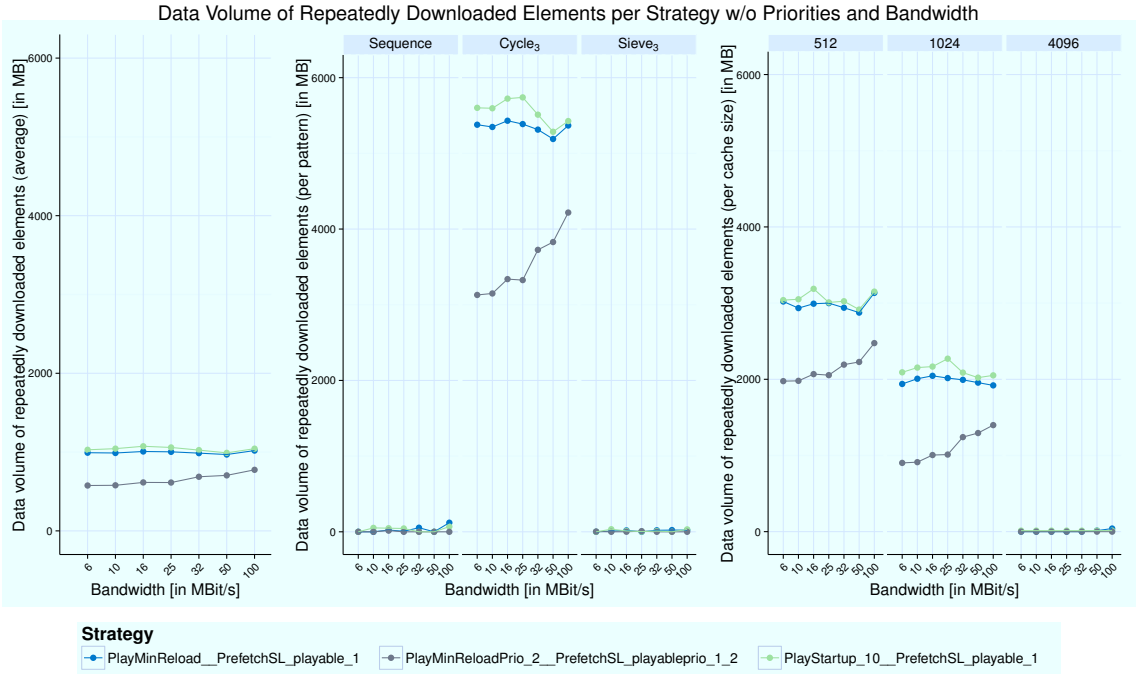


Figure 7.53.: Evaluation of the patterns (selected results, priority-based strategy and start time) - data volume of repeatedly downloaded elements for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

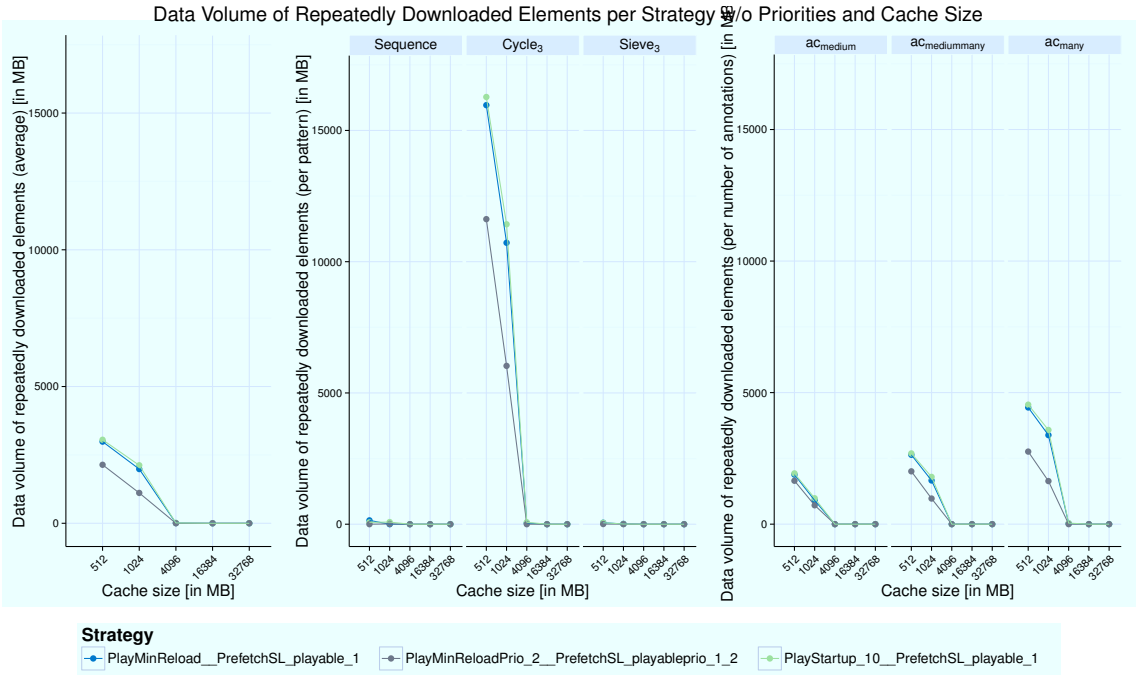


Figure 7.54.: Evaluation of the patterns (selected results, priority-based strategy and start time) - data volume of repeatedly downloaded elements for different cache sizes: average for the whole test (left), results grouped by pattern (center), and results grouped by bandwidth (right).

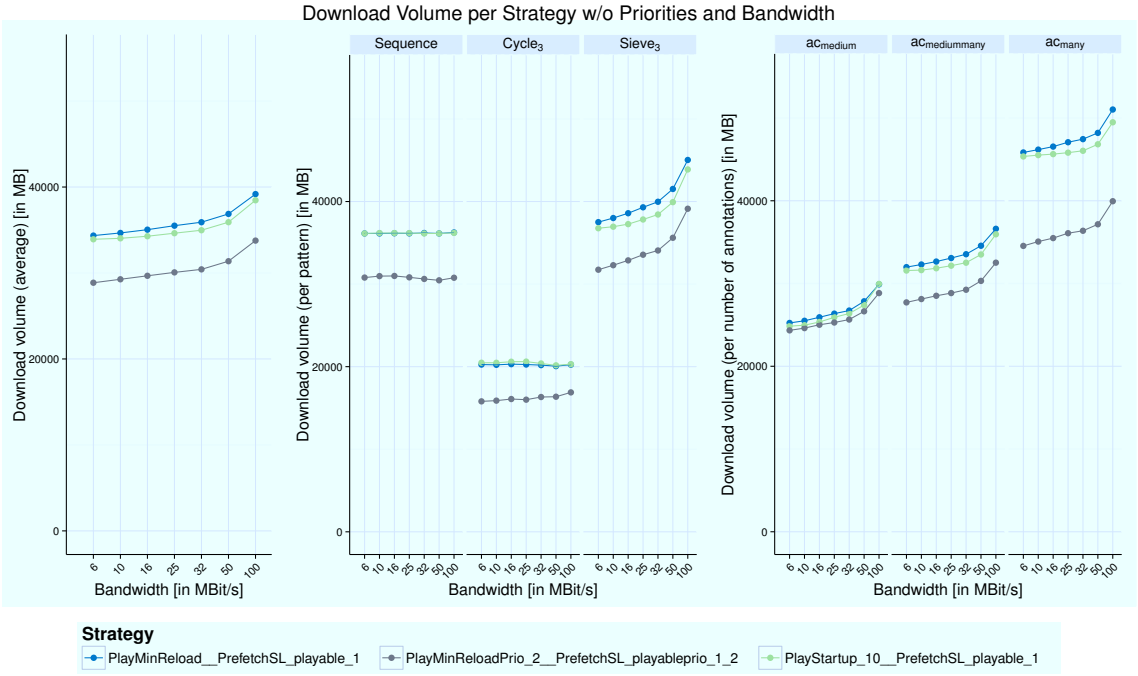


Figure 7.55.: Evaluation of the patterns (selected results, priority-based strategy and start time) - download volume for different bandwidths: average for the whole test (left), results grouped by pattern (center), and results grouped by cache size (right).

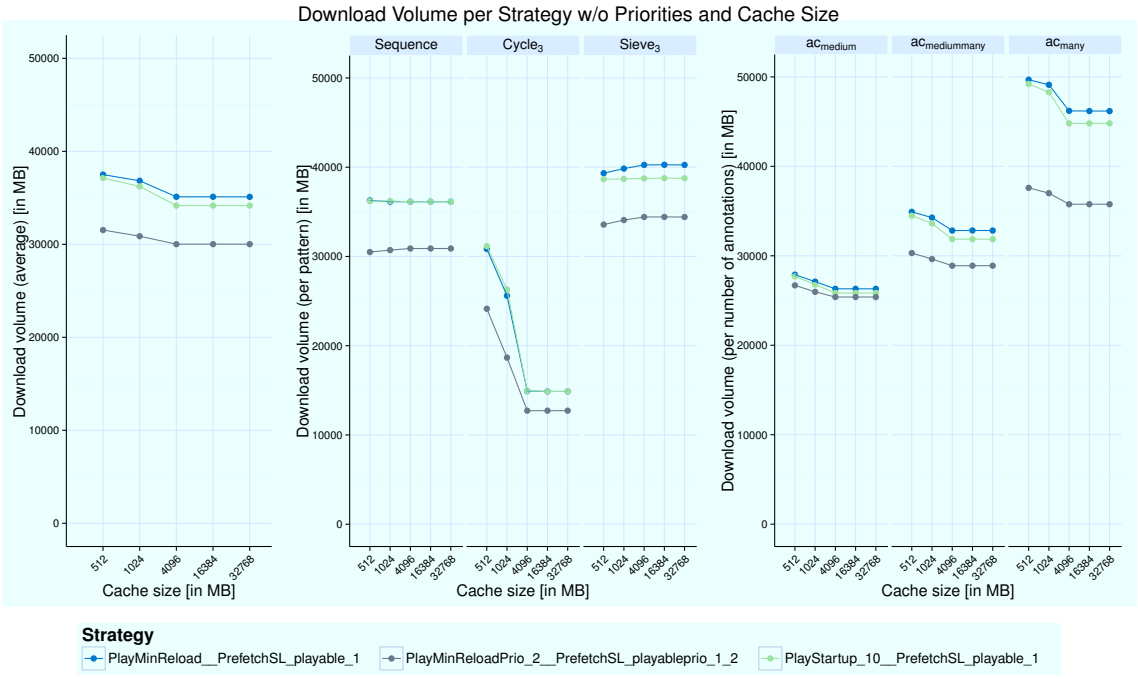


Figure 7.56.: Evaluation of the patterns (selected results, priority-based strategy and start time) - download volume for different cache sizes: average for the whole test (left), results grouped by pattern (center left), results grouped by used probabilities (center right), and results grouped by bandwidth (right).

- The differences between the results for the frames to wait for these two strategies are the bigger the higher the bandwidth is independent from the cache size.
- Both strategies achieve about the same average values for small bandwidths for the frame to wait at the beginning of a scene.
- The waiting time at the beginning of scenes is always smaller for the priority-based strategy than for the PLAY_MIN_REL strategy and a pre-fetch depth of one scene, whereby the differences become smaller the larger the regarded bandwidth is.
- The differences between the results for the waiting time at the beginning of scenes increases with an increasing number of annotations per scene for these two strategies.
- For the priority-based strategy, the value for the P_{sum} metric is smaller or equal to the PLAY_MIN_REL strategy and a pre-fetch depth of one scene.
- The differences between the results for the pauses during scenes decreases with an increasing bandwidth and/or cache size.
- The number of skipped frames ($A_{skipped}$) is the higher the smaller the cache size and the bandwidth are; it decreases with higher cache sizes and bandwidths.
- The results for the data volume of downloaded but not watched elements ($DL_{not\ watched}$) are not always better for the priority-based strategy than for the PLAY_MIN_REL strategy and a pre-fetch depth of one scene.
- The results for the data volume of repeatedly downloaded elements ($RDLV$) show that the results for the priority-based strategy are better or equal than for the PLAY_MIN_REL strategy and a pre-fetch depth of one scene.
- The results for the ($RDLV$) metric are significantly better for the cycle_3 pattern and small cache sizes using the priority-based strategy than the strategy with a PLAY_MIN_REL strategy and a pre-fetch depth of one scene.
- The overall download volume (DLV) is smaller for the priority-based strategy than for the strategy with a PLAY_MIN_REL strategy and a pre-fetch depth of one scene.
- The number of annotations has a large influence on the difference between both strategies regarding the download volume; it is increasing with an increasing number of annotations.

7.6. Summary

An evaluation of different combinations of the strategies for the determination of the start frame for playback (PLAY_SCENE, PLAY_MIN_REL, and PLAY_STARTUP), for the prefetch (PREFETCH_SL), and for the deleting (DELETE_SD, DELETE_LRU, DELETE_D_PROB, and DELETE_PRIO) revealed the following findings with regard to the tested combinations of bandwidth, cache size, count of annotations, annotation priorities, number of paths and path probabilities:

- The start time as well as the pre-fetch strategy have huge influence on the measured metrics while the delete strategies have only little influence.

- Pauses during playback can be avoided without pre-fetching whole scenes if the cache is large enough to hold the scene up to the point in time where the playback is started according to the PLAY_MIN_REL strategy.
- A pre-fetch depth of one scene is enough to reduce waiting times at the beginning of scenes significantly.
- A pre-fetch of one scene or more always leads to an increased download volume in branching patterns. In contrast, no increase occurs in linear and cyclic patterns when suitable cache sizes are used, because no elements are downloaded repeatedly or downloaded but not watched.
- The delete strategy should be able to decide on elements to delete with knowledge of the whole scene graph. The repeated download of elements can be reduced and the required overall download volume can be minimized.
- A recommendation for a combination of the individual strategies can be given for different user requirements and environmental settings.
- The results of the proposed strategies are stable with regard to the number of annotations and the pattern width with varying path probabilities.
- The proposed strategies confirmed the results from the pattern-based tests in the user generated scenarios.
- The usage of priorities for annotations can help to reduce download volume and waiting times if not all annotations are necessary during playback. This makes the priority-based strategy highly applicable for mobile devices.

8. Conclusion and Future Work

The SIVA (Simple Interactive Video Authoring) Suite was designed and implemented to realize annotated interactive non-linear videos. This software suite consists of an XML schema definition which determines the structure of the data exchange format and realizes our video model combining interactivity, non-linearity, and annotations. Both, the authoring tool (SIVA Producer), and the playback units (SIVA Players) are implementing this model. The SIVA Producer has a scene graph editor, different annotation editors, and a table of contents editor, as well as several other useful functions. The SIVA Players interpret the XML file exported by the SIVA Producer. They display the video and the annotations in an video player which among other features provides hotspots and an extended timeline. Thereby, they provide functions exceeding those of conventional players. During the design phase of our software, we integrated the lessons learned from related work, for example the usage of an event-based timing model in our XML format or the combination of different authoring paradigms in the SIVA Producer.

The second part of this work proposes and evaluates algorithms and strategies for download and cache management in annotated interactive non-linear videos. An analysis of related work from different areas revealed basic ideas for our algorithms and strategies. We implemented algorithms for the scheduling of scenes, `SortSceneLinear` and `GetStartFrame`, and for parts of the scene graph, `SortVideoLinear`, in one sequence at full bandwidth. We furthermore developed strategies for the determination of a start frame for playback (PLAY_SCENE, PLAY_MIN_REL, and PLAY_STARTUP), for the pre-fetch of elements into (PREFETCH_SL), and for the deleting (DELETE_SD, DELETE_LRU, DELETE_D_PROB, and DELETE_PRIO) of elements from the cache. These strategies are combined to sets of strategies which were then tested in different combinations of bandwidth, cache size, count of annotations, annotation priorities, number of paths and path probabilities. The tests revealed that the calculation of the start frame for playback without pauses from which on the rest of the scene can be downloaded during playback shows advantages in the selection of the start point for playback as well as a download portion compared to choosing the last frame of a scene or a frame after a fixed time. Furthermore, the strategies which take the further course of the video into account show advantages compared to traditional cache replacement algorithms. With our algorithms and strategies, the quality of experience during playback can be improoved while the additional download volume is minimized.

Future work concerning the SIVA Suite is possible in different areas. Possible extensions for the SIVA Producer are an enhanced usability by defining different video profiles, for example for virtual tours or e-learning. Pre-defined patterns and limited functions may allow the author to finish her/his project faster. Furthermore, different forms of user support can be integrated to help the user with different tasks. Algorithms for object or text detection [Pei10; MPK10] to simplify the creation of annotations or to provide keywords extracted from image annotations can be added. Shot and scene detection algorithms [ZMK14; Zwi12] accelerate the creation of the scene graph providing scenes from longer videos with only little work input.

The usability could further be enhanced by integrating a player preview in the authoring tool which allows to watch the project without exporting it first. Collaboration and logging are possible extensions for the SIVA Players. Concerning the collaboration feature, more types of annotations should be possible. Furthermore not only extending the annotations of existing videos, but changing the structure of videos are future challenges. Useful ideas for that feature can be found in the works of Mirri et al. [Mir+11], Konstantinidis, Tsiatsos, and Pomportsis [KTP09], and Singh et al. [Sin+11], whose implementations need to be adapted for our model. The collection of logging data, especially in e-learning scenarios, may be used to identify learning paths for certain groups of learners. Frequently used paths could then be improved either based on the learning times or on collaborative elements added by the different learners. With a better support of download and cache management in HTML5, the scenes of often selected paths could be pre-fetched to decrease loading times. Therewith, the quality of experience during playback could be increased due to the shorter waiting times.

Future work concerning download and cache management could include topics from different areas. In this work we used the click of a user on a button as decision criterion at a forks. The described model for annotated interactive non-linear videos could be extended and refined for more complex, rule-based, decision criteria [Wac13]. Decisions may be based on local and global variables which result from previous clicks on buttons, quiz results, the history of a user, viewed annotations, logging, counters, and timestamps. The combination of these variables requires to deal with problems like the initialization of the variables and the solving of the expressions. Furthermore, it is possible that not all variable assignments are covered by the author or a combination of variables results in a dead end in the scene graph. Mechanisms need to be found which enable the viewers to resume the playback or reach the end of the video in any case. Another area for possible future research concerning our algorithms and strategies is the collaboration function in the player. The current implementation takes viewer annotations into account only after a scene change. A higher quality of experience can be achieved if new annotations are displayed to other users immediately after their creation. This requires a rescheduling for the currently watched scene in any case. Algorithms may add buffers for additional downloads in order to avoid stalling events during playback and an immediate response to viewer interaction. Further improvements of our algorithms and strategies may be possible, for example if video and audio files are not treated as a whole download block but as single downloadable frames or portions of audio files. This requires an adaptation of our algorithms and strategies as well. In addition, we did not consider a more interactive and “jumpy” user in this work. We tested our algorithms and strategies for user interaction at the end of a scene. Further tests where the viewer jumps within one scene or by using the keyword search or the table of contents might be interesting as well. Being implemented in Java, the simulation framework could be integrated in our SIVA Producer to give the author some hints on how the video project should be exported in order to provide a good quality of experience to the viewers. Performance issues during playback could be avoided using this function. Furthermore, the performance during playback and thus the user experience could be improved with content adaptation, like spatial and/or quality scaling, as suggested by Avramova et al. [Avr+11b]. The most serious challenge in future work is the implementation of our strategies into our HTML5 player. HTML5’s `prefetch` attribute of the `<video>` element underwent several changes in the standardization process. The current version of the standard as well as the current implementations in the browsers do not allow the definition of the amount of data to be pre-fetched. It is furthermore not possible to keep elements in the browser cache for later reuse. The whole download and cache management, as described and evaluated in this work, could be implemented for our player in JavaScript.

This would result in a complete redevelopment of the functionality now provided by the `prefetch` attribute and its implementation in the browsers.

A. List of Symbols

Table A.1.: List of symbols.

Symbol	Explanation
$A_{skipped}$	Metric for the number of frames with skipped annotations
B	Symbol for the cache size function
BW	Symbol for the bandwidth function
$DL_{not\ watched}$	Metric for data volume of downloaded but not watched elements
DLV	Metric for the overall download volume
H	Symbol for the absolute frequency
P	Symbol for a part of a scene
$P1$	Part of a scene that has to be downloaded for playback without breaks
$P2$	Part of a scene that can be downloaded during playback without annotations with low priorities
$P3$	Part of a scene that can be downloaded during playback
P_{sum}	Metric for the number of pauses during scenes
$RDLV$	Metric for data volume of repeatedly downloaded elements
$\mathcal{S}_{p_x}$	Valid schedule for one scene p_x
WF_{start}	Metric for the number of frames to wait at the beginning of a scene
WT_{start}	Metric for the waiting time at the beginning of a scene
a	Symbol for an annotation
a_o	The o-th annotation of a video
ac	Number of annotations in a scene
ac_{few}	Few annotations in a scene
$ac_{fewmedium}$	Few to medium annotations in a scene
ac_{medium}	Medium annotations in a scene
$ac_{mediummany}$	Medium to many annotations in a scene
ac_{many}	Many annotations in a scene
b	Symbol for download specifications
b_1	Part of a scene which is unlocked for download in PREFETCH_SL
b_2	Distance for that the scenes are unlocked for download in PREFETCH_SL
c	Symbol for constant values
c_B	Constant for the cache size
c_{BW}	Constant for the bandwidth
$c_{BW^{device}}$	Constant for the bandwidth of a device
c_r	Constant for the frame rate (normal speed)
$c_{r_{SFW}}$	Constant for the frame rate (slow forward)
$c_{r_{SBW}}$	Constant for the frame rate (slow rewind)
$c_{r_{FFW}}$	Constant for the frame rate (fast-forward)
$c_{r_{FBW}}$	Constant for the frame rate (fast rewind)

dim	Symbol for the function returning the length of a tuple
dl	Symbol for the download duration function
dur	Duration of a scene
dur_{short}	Short scene duration
dur_{medium}	Medium scene duration
dur_{long}	Long scene duration
dur_{combi}	Combination of scene durations
e	Symbol for a (downloadable) element
e_i	The ith element of set $\mathcal{E}_\mathcal{V}$
f	Symbol for a frame
$f_{i,m}$	The mth frame of scene i
g	Some function
g_B	Function for varying cache sizes
$g_{BW^{device}}$	Function for varying bandwidth per device
g_i	Index function for the i-th delete index
h	Symbol for the relative frequency
i	Index
j	Index
j_i	Last frame index of scene i
k	Index
l	Symbol for the duration function
m	Index/frame index from which a scene can be played without reloads
max	Symbol for the function returning the maximum of a result set
$mean$	Symbol for the function returning the mean value of a result set
$median$	Symbol for the function returning the median of a result set
min	Symbol for the function returning the minimum of a result set
n	Index
o	Index
p	Symbol for a scene
p_i	Scene i in set $\mathcal{P}_\mathcal{V}$
$\widehat{p}_x^f$	Tuple of frames of scene p_x
$\widehat{p}_x^a$	Tuple of annotations of scene p_x
$\widehat{p}_x^e$	Tuple of elements of scene p_x
$prob$	Function returning a probability for a selected path
$prob_{best}$	Function returning the highest probability for the selected path
$prob_{bestavg}$	Function returning the probability between highest and average for the selected path
$prob_{avg}$	Function returning the average probability for the selected path
$prob_{worstavg}$	Function returning the probability between average and lowest for the selected path
$prob_{worst}$	Function returning the lowest probability for the selected path
q	Symbol for the function returning the priority of an element
r	Symbol for the frame rate function
s	Symbol for the function returning the size of an element
sd	Symbol for the function returning the standard deviation of a result set.
$size$	Resolution for a scene
$size_{small}$	Small size of the scene
$size_{medium}$	Medium size of the scene

$size_{large}$	Large size of the scene
t	Symbol for the time
t_i	The ith point in time
$t_{current}$	The current point in time
$t_{startup}$	The start-up time
u	Variable for the choice at a fork
v	Variable for the width of a pattern
w	Symbol for a button
w_j	The jth button
x	Index
y	Highest index of a scene
z	Symbol for the definition of a download portion
z_{SL}	Symbol for the definition of the download portion of a scene in PREFETCH_SL
z_{FF}	Symbol for the definition of the download depth in PREFETCH_FF
α	Symbol for the content of an annotation
α_o	The content of the oth annotation in set $\mathcal{A}_\mathcal{V}$
γ	Amount of a scene in the prefetch strategies
δ	Symbol for the optimization function
Δ	Function for the temporal distance
ϵ	Symbol for the end of the video
Λ	Priority of an annotation
μ	Weight factor
π	Symbol for the projection function
π_i	Symbol for the projection function on the i-th element of a tuple
$\pi_{i,j}$	Symbol for the projection function on the i-th th j-th element of a tuple
σ	Symbol for the start of the video
τ	Function for a waiting time
ϕ	Timeout at a fork
$\mathbb{N}$	Set of natural numbers
$\mathbb{N}^+$	Set of positive natural numbers (without zero)
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers (without zero)
$\mathcal{A}_\mathcal{V}$	Set of annotations of $\mathcal{V}$
$\mathcal{A}_{p_x}$	Set of annotations of scene p_x
$\mathcal{DX}_\mathcal{V}$	Set of downloaded elements
$\mathcal{DXN}_\mathcal{V}$	Set of downloaded but not watched elements
$\mathcal{DXR}_\mathcal{V}$	Set of repeatedly downloaded elements
$\mathcal{E}_\mathcal{V}$	Set of (downloadable) elements of $\mathcal{V}$
$\mathcal{E}_{p_x}$	Set of (downloadable) elements of scene p_x
$\mathcal{F}_\mathcal{V}$	Set of frames of $\mathcal{V}$
$\mathcal{N}_\mathcal{V}$	Set of transitions of $\mathcal{V}$
$\mathcal{P}_\mathcal{V}$	Set of scenes of $\mathcal{V}$
$\mathcal{P}_{succ(p_i)}$	Set of successor scenes of scene p_i
$\mathcal{P}ause_\mathcal{V}$	Set of pauses during playback
$\mathcal{P}ath_\mathcal{V}$	Tuple of elements of a path
$\mathcal{X}$	Random set of single elements

$\mathcal{X}^k$	Random set of k-tuples
$\mathcal{V}$	Symbol for an interactive non-linear video
$\exists$	Existential quantifier (“there exists”)
$\exists!$	Existential quantifier (“there exists exactly one”)
$\nexists$	Existential quantifier (“there does not exist”)
$\forall$	Universal quantifier (“for all”)
$\cap$	Intersection of sets
$\cup$	Union of sets
$\lvert\circ\rvert$	Cardinality of a set

B. Standards and Models

Standards/Models for Hypermedia Applications

Source/cite	*Scope*	*Main medium/ video scenes*	*Non-linear structure (defined by author)/ structure of the model*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Additional information*	*Temporal model/ description*	*Spatial model/ description*	*Language*	*Model*	*Based on*
HyTime (Hypermedia/ Time-based structuring language) [Gol91; NKN91; Erf93]	"HyTime is about addressing, linking, and alignment. Addressing deals with identifying a certain amount of information [...]. The linking features allow to create links between parts of information [...] alignment supports placement of pieces of information within finite coordinate systems."	images, text, audio files, video	"links between parts of information" build a graph structure	jumps to frames, no influence on order of scenes	no/not described (only VCR)	images, text, audio files, video	"all issues of specifying temporal constraints within multimedia documents can be represented [...] require application-defined element types or marker functions."	"alignment supports placement of pieces of information within finite coordinate systems"	SGML	no/not described	not described
Amsterdam Hypermedia Model (AHM) [HBR94; HB97; HWB97]	"general framework that can be used to describe the basic constructs and action that are common to a wide range of hypermedia systems", adds "high-level presentation, attributes and link context to the Dexter model [HS94]"	"collection of media items" (video, audio files, image, text)	graph structure defined by hyperlinks	no/not described	"Interactions, including navigation, [...] jumping to related information.", hyperlinks, menues and other structures, hotspots not described	"collection of media items" (video, audio files, image, text)	"temporal dependencies among media items, possibly stored at different sites.", temporal relations, components which contain media files, synchronization arcs between components	"logical connections among items, including the grouping of items to be displayed together and the specification of links among these groupings. Layout specifications state where screen-based media are to be sized and placed, either in relation to each other or to the presentation as a whole."	only model	"general hypermedia model"	Dexter Model [HS94] and CMIF multimedia document model

MHEG-5 [Ech+98]	"[...] encoding format for multimedia applications independently of service paradigms and network protocols."	bitmap, polylines, ellipses, text, audiovisual data	graph structure defined by hyperlinks	jumps in timeline, but no influence on order of scenes	"Interaction can be performed via graphic elements", interactive thumbnails, text links	bitmap, polylines, ellipses, text, audiovisual data	"a set of scenes, which contain objects common to all scenes. A scene supports the spatially and temporally coordinated presentation of audiovisual content"		object-based declarative programming language	object-oriented model	not described
Celentano and Gaggi [CG00; GC05]	"model for describing the synchronization between several media delivered over a network in a Web-based environment"	videos	"link structure in and between hypermedia documents"	jumps triggered by hyperlinks, no influence on order of scenes	hyperlinks, but no hotspots	video, audio files, text, images	"Synchronization is achieved with a set of relationships among the components of a multimedia presentation"	"spatial positioning is obtained by channels definitions"	no/not described	synchronization model	not described

Table B.1.: Standards/models for hypermedia applications.

Standards/Models for Hypervideos

Source/cite	*Scope*	*Main medium/ video scenes*	*Non-linear structure (defined by author)/ structure of the model*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Additional information*	*Temporal model/ description*	*Spatial model/ description*	*Language*	*Model*	*Based on*
Generalized Hyper-Video System (GHVS) model [Hun97]	"specify the hyperlink from one frame to another. [...] physical data independence, the ability to compose arbitrarily complex presentations of hypervideos, completeness in expressibility, and simplicity."	video scenes	graph, defined by video-to-video hyperlinks	jumps to other frames triggered by hotspots, no influence on order of scenes	rectangled hotspots hyperlinks/ jumps between scenes and to audio files, sound, and images	video, image, audio files, sound	no description	areas (rectangles) defined with left-up and right-down coordinates	GHVS language	"(GHVS) model"	PRES model [WKD96]

Logical Hypervideo Data Model (LHVDM) [JE98]	"In addition to multilevel video abstractions, the model is capable of representing video entities that users are interested in [...] and their semantic associations with other logical video abstractions, including hot objects themselves."	videos	graph structure defined by inter-video hyperlinks	intra-video hyperlinks, no influence on order of scenes	hotspots (hot objects) hyperlinks to same or other video	content generated from video (audio files, image frame)	"time intervals are defined as live time intervals (LTIs) of a hot object.", "Spatio-temporal constraints describe the dynamic features of HOs which are unique to video data."	"Spatial relations of hot objects can either be quantitative or qualitative.", "geometric or spatial information, such as boundary, shape, position and orientation"	"video query language based on the LHVDM model"	"Logical Hypervideo Data Model"	not described
Chambel and Guimaraes [CGa02]	"hypervideo model is based on the hypermedia model of the Web, extended with additional concepts required to support the temporal dimension of hyperlinking in dynamic media, such as video."	video	graph (hyperlinks between video and other media): different links, table of contents, various maps	jumps to points on the timeline as defined in a video index, no influence on order of scenes	hyperlinks, maps, table of contents; hotspots: "link anchors can be spatially scattered in the pages and images"	images, text,	"Temporal links are only dependent on time conditions. A link can be established for a time interval. Different links can be established from different sub-videos"	"Spatial links are only dependent on space conditions, making it possible to establish links from different spatial regions of the video. These are always active, while the video is playing"	"HTIMEL (our extended language for hypervideo construction)"	hypervideo model	[CCG01]
Component-based Hypervideo Model (CHM) [SAP11]	"high level representation of hypervideos that intends to provide a general and dedicated hypervideo data model"; spatial, temporal and event-based models	video	link structure between videos	jumps to points on timeline, in map, in history or table of contents, no influence on order of scenes	"data readers, enrichment content viewers, video players, [rectangled] hotspots, timelines, maps and tables of contents"	text, video, audio files, rich text, audio files	"timeline-based model. The explicit time scale of document components is defined by [...] a virtual time reference attached to a video playback component or to the global document."	"Derived from the SMIL spatial model, the CHM spatial model is intended to accommodate the implementation platform specificities."	no/not described	"CHM"	spatial model based on SMIL

Table B.2.: Standards/models for hypervideos.

Standards/Models for Interactive Multimedia Presentations

Source/cite	*Scope*	*Medium*	*Non-linear structure (defined by author)/ structure of the model*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Temporal model/ description*	*Spatial model/ description*	*Language*	*Model*	*Based on*
SMIL 3.0 [BR08; W3C12]	"XML-based language that allows authors to write interactive multimedia presentations. [...] describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen."	images, text, audio files, video, animation, textstream	graph structure defined by links between elements	jumps in one multimedia presentation	choice elements can be defined with SMIL elements, hotspots can be defined in different shapes, trigger action	"using the elements and attributes defined in the 19 timing modules, time can be integrated into any XML language", definition of start and end time, duration, persistence, repetition, accuracy	"relative placement of (multiple) media objects, but not the internal formatting of any of the individual objects"	synchronized multimedia integration language	no/not described	AHM, CMIF
NCL 3.0 [Int11; Sil^{+}04; Tel11]	"describe the temporal behaviour of a multimedia presentation, associate hyperlinks (user interaction) with media objects, define alternatives for presentation (adaptation), and describe the layout of the presentation on multiple devices."	image objects, video objects, audio objects, text objects, imperative objects, other declarative objects	no/not described	no/not described	jumps, "switch element definition (content alternatives)"; "the descriptorSwitch element definition (presentation alternatives)", hotspots not described	"definition of anchors representing temporal portions, through begin, end and dur (as in SMIL)"	"anchors representing spatial portions, through the coords attribute (as in XHTML),"	XML-based	NCM model	NCL inherited several modules from SMIL

Table B.3.: Standards/models for interactive multimedia presentations.

Standards/Models for Multimedia Presentations

Source/cite	*Scope*	*Medium*	*Non-linear structure (defined by author)/ structure of the model*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Temporal model/ description*	*Spatial model/ description*	*Language*	*Model*	*Based on*
CWI Multimedia Interchange Format (CMIF) [BRL91]	"document structure for describing transportable, dynamic multimedia documents", "describe the temporal and structural relationships that exist in multimedia documents", "synchronization channels, event descriptors, data descriptors, data blocks and synchronization arcs"	media blocks: "sound clips, video segments, text blocks, graphics images, etc"	"document tree that is used to encode the hierarchical and peer relationships among document events."	no/not described	no/not described	temporal relationships between media blocks	spatial relationships between media blocks	no/not described	no/not described	AMF
Madeus [LSI96]	"an interval based temporal model and constraints which provide a basis for the management of the consistency of multimedia documents"	text, pictures, graphics, video, audio files	no/not described	no/not described	no/not described	temporal structure is defined as a set of temporal relations between basic media objects and composite objects	allocation of media channels	no/not described	"interval based temporal model"	[All83]

Layered Multimedia Data Model (LMDM) [SW94]	"model for specification of MM data and MM compositions"	audio files, video, image, text	event structure, temporal structure, definition of hyperlinks possible	jumps in timeline, but no influence on order of scenes	"creating hyperlinks, adding persistent bookmarks or trails, or developing their own navigation tools", hotspots not described	"An MM event is built from one or more MM objects from the DDL [Data Definition Layer], each of which has been assigned a temporal component, and which have been temporally aligned. [...] application of the MM-event calculus. The event calculus provides operators for sequencing and temporally overlaying the occurrences of the objects in the event."	"The Data Presentation Layer (DPL) provides a description of how data is to be communicated to the user. [...] presentation dependencies between objects or events, spatial layout, output format and user interface elements such as windows or icons."	"DML contains a symbolic language [...] as well as an event calculus", "The CL provides a scripting language"	"layered multimedia data model"	not described
PREMO (Presentation Environment for Multimedia Objects) [HRL96a], [HRL96b]	"Premo is a presentation environment that aims to provide a standard programming environment [...] targets multimedia presentation [...] High-level virtual reality environments, which mix real-time 3D rendering techniques with sound, video, or even tactile feedback"	text, video, audio files, images, animated graphics	no/not described	no/not described	interaction possible, hotspots not described	"Clock objects provide a unified interface to the system's view of a real-time clock", "event-based synchronization model in which each synchronizable object progresses autonomously along an internal, one-dimensional coordinate space"	"The properties of output primitives specify their geometry and appearance. These properties are currently classified as spatial, visual, aural, tactile, textual, and identification, although debate on the exact details continues."	"The formalism adopted for Premo's specification builds on the Z and the 0bject-Z languages [...]"	"Premo object model"	"OMG model", [Car93; ISO92]
Adali et al. [ASS99; ASS00]	"[...] algebra for querying multimedia presentation databases", "[...] operate on trees whose branches reflect different possible playouts of a family of presentations"	"relational table, text document, image, video segment, audio segment, web page, etc."	tree structure	no/not described	object, node and path selection conditions, hotspots not described	"st(o), et(o) called temporal variables (denoting the start and end time of object o)"	"ulc(o), urc(o), llc(o), lrc(o) called spatial variables (denoting the upper left corner, upper right corner, etc. of object o)"	only model	algebraic model	[LO96], [OHK96]

Adiba and Zechinelli-Martini [AZM99]	"[...] a special emphasis on spatial aspects and we provide both qualitative and quantitative relations to compose and query multimedia presentations. [...] presentations can be specified, stored as database objects, queried and executed."	texts, images, video and audio data	graph?	no/not described	no/not described	"temporal dimension, the Temporal Shadow (TS) of an object is defined by δ that represents the delay during which the object is ready to be displayed but not yet 'perceptible' and by d, the effective duration of the presentation"	"The Spatial Shadow (SS) [...] describes the position of each object (x, y) and its size (length and width) (dx, dy). Thus, the SS describes the spatial attributes of an object in a given presentation. [...] SS is similar to the notion of Minimum Bounding Rectangle (MBR)."	"representation is independent of any description language and media type"	database model	extends O2, take advantage of OQL (Object Query Language)
Assimakopoulos [Ass99]	"Temporal interval relations [...] need to be analyzed to ensure that there is no conflict among resources."	multimedia resources	"complete graph.: contains user edges and derived edges with possible cycles and possible conflicts."	no/not described	no/not described	"domain of interval temporal relations is analyzed and a directed graph to compute the relations of multimedia objects is used"	"The work discussed in Allen (1983) only states temporal interval relations. We found that these relations can be generalized for spatial modelling."	no/not described	computation model	[All83]
Procedural Markup Language (PML) [Ram+99]	"decouples content and presentation. It lets users specify the knowledge structures, underlying physical media, and relationships between them using cognitive media roles."	"text, graphics, animations, video clips, and sound files"	graph	no jumps in one video, "adjust itself [(the presentation)] in response to a user's goals."	hyperlinks to other contents, no hotspots described	not described	not described	XML, specified by DTD	no/not described	not described

MPGS [BFS00]	"[...] supports the specification of constraints among multimedia objects and the generation of multimedia presentations according to the specified constraints"	"text, image, video, sound, and graphic"	no/not described	no/not described	no/not described	"A static object does not have a temporal dimension. By contrast, a dynamic object has an implicit temporal dimension.", temporal constraints	"[...]rectangle as the minimum bounding rectangle (mbr) of the object. [...] defining its height, width, and the spatial distance between its upper left corner and the upper left corner of the monitor object.", spatial constraints	no/not described	"multimedia presentation model"	[All83]
ZYX [BK01]	"multimedia document model for reuse and adaptation of multimedia content"	audio files, video, image, text	temporal and spatial model	"navigational/ decision interactions and design interactions"	"genericLink element that allows us to specify the transition from the document to an arbitrary link target" (nit interactive), "elements hotspot and hypertext define fine-grained interactive visual areas in images and text"	"the model offers the primitives seq, par, loop, and delay to specify temporal interval relationships"	"absolute positioning", usage of spatial projectors	formal definition	"ZYX models describes a multimedia document by means of a tree"	not described
Deng et al. [Den+02a]	"a new approach for the modeling of reusable and adaptable multimedia content"	media objects	Petri net	no/not described	no/not described	"They are three types of temporal operator elements: [...] Ts [...] bound the presentation media element in sequence. [...] Tp [...] render the presentation media elements in parallel. [...] Te [...] a exclusive of transitions."	"A spatial operator element applies the presentation place P with specific absolute/relative position parameter: x-axis index, y-axis index, z-index, height, and width."	not described	content model	Petri net

XiMPF: eXtensible Interactive Multimedia Presentation Format [VA+04]	"a generic publication framework that features a simple data model and a flexible filter architecture"	"individually identifiable asset such as a video or audio clip, an image, or a textual asset."	more or less the same structure as MPEG-21 DIDL	no/not described	different types, open new information, font selection,	"temporal layout of the presentation" (no further description)	"spatial layout of the presentation" (no further description)"	associated file format	data model	MPEG-21 Digital Item Declaration Language (DIDL)
Scherp and Boll [SB05]	"abstract multimedia content model that embeds the central characteristics of today's multimedia presentation formats: the definition of the temporal and spatial layout as well as the interaction possibilities."	images, text, audio files, video	internal links, external links	jumps in one multimedia presentation to defined point, no influence on order of scenes	choice/ control functions not described, hotspots can be defined, but no further description in paper	"abstract content model provides a set of temporal composition elements: The Parallel element [...] Temporal Selector [...] composition element Sequential [...] composition element Delay [...], all temporal relations as defined by Allen [14] can be modeled."	"we decided for our abstract spatial model in favor of relative positioning"	no/not described	"abstract multimedia content model"	not described

Table B.4.: Standards/models for multimedia presentations.

C. Authoring Tools

Authoring Tools for Non-linear Videos

Source/cite	*Scope*	*Main medium/video scenes*	*Non-linear structure (defined by author)*	*Influence on order of scenes (before playback)*	*Choice elements/ hotspots*	*Additional information*	*Language*	*Patterns/editors*
Riva Producer Enterprise [mem13]	"splitting the videos into small information units and by the use of annotations for each unit you get a video-based database"	videos, cut into scenes	yes	no/not described	yes, hotspots which lead to another scene	graphics and buttons in the video	own XML-format	no screenshot of the software available
XIMPEL [Bhi+10]	"create interactive media applications" , "create their own storylines"	video scenes	yes, graph	no/not described	different hotspots at the same time, invoke next scene	text and image, questions with evaluation	self-defined XML-format	XML-editor (no GUI available)
YouTube Video Annotations [You13]	"add interactive commentary to your videos": "add background information about the video", "create stories with multiple possibilities", "link to related YouTube videos, channels, or search results from within a video"	linear YouTube videos	yes, graph (not visible, created by links)	no/not described	different types of hotspots, link to another YouTube video or homepage	speech bubble, note, title (text), spotlight (hotspot), label	not described	parallel timeline, WYSIWYG-video preview, form to define annotation

Table C.1.: Authoring tools for non-linear videos.

Authoring Tools for Interactive Videos

Source/cite	*Scope*	*Main medium/ video scenes*	*Alternative playback paths (jumps)*	*Choice elements*	*Additional information*	*Hotspots*	*Language*	*Patterns/editors*
Zodiac [Chi+00]	"basis for interactive document version navigation, accurate shot/scene detection and simplified video object annotation authoring"	sequential linear videos or parts of videos	not described	no/not described	text, image, video audio files	used to show additional information with an object in the video	not described	timeline with thumbnail-preview
HyStream System [Bea+02]	link creation for continuous media, approach enriches hypermedia content with additional metadata	one linear video	no/not described	no/not described	presentations with slides and labeled links (with start- and end-point)	no/not described	RDF	video preview with editing function
LazyMedia [HL06]	"fast, flexible and personalized video authoring and sharing"	scenes combined to one linear video	alternative playback paths (jumps) based on video chapters	video chapters	images, text, audio files	no/not described	LMPF file	library tree, library view (preview), player, property list, timeline
Chang et al. [CHS07; CHC08]	create educational games easily	video files which are divided into scenario components	"buttons also provide players options to switch to other video segments"	buttons that appear in the video	images, links to websites	no/not described	no/not described	tree structure, information areas, time sliders, video area
HyLive [HKH08]	"interactive live television with hypervideo structures"	linear video	no/not described	no/not described	elements for interactions like voting and hypervideo links which refer to additional information about content	rectangled areas which open additional information	not described	view consisting of three parts, more details from left to right, tree-based

Composer [LGaMDRCGS08]	"authoring tool to help creating interactive TV programs"	video/media	no/not described	yes	"any specific media content (video, audio files, text, imperative code etc.)"	no/not described	Nested Context Language 3.0	textual view, structural view, layout view, temporal view
Chen et al. [Che+09a]	"combine the video-based course materials and game elements with an integrated learning platform called"	sequential linear videos	no/not described	no, but elements that have to be used to access the next scene	games, images, text	no	SCORM	different text and media areas
SeViAnno [Cao+10]	interactive semantization of multimedia	one linear video	yes, navigation by annotations	no/not described	semantic annotations, place annotations	no/not described	MPEG-7	video preview with editing function
Räck et al. [RSA10]	"interaction with video data on TVs, Mobiles and the Web [...] deliver value-added information, links and advertisements on-demand and in a personalized way"	video	no/not described	no/not described	image, text	used to show additional information with an object in the video	Object Definition Language (ODL)	web-based forms for object identification, tracking and linking
ADIVI Production Kit [Inn11]	"hypervideo- and rich-media-application which enables you to create interactive videos. [...] communicate information without any media discontinuity. [...] merges the advantages of website and video in one single media."	one linear video	no/not described	no/not described	"different multimedia information like additional videos, documents, pictures etc."	rectangles or circles which invoke additional information	not described	WYSIWYG-editor, timeline, controls, sidebar for creation of hotspots, input fields for annotation-content
Quicktvpro [Bel12]	"powerful editor and extensive range of tools allows you to create your interactive video effects exactly as you want them"	linear video	jumps in the video to beginning of chapters	no/not described	voting and polling, links to sales web pages, images, text, shapes, SWF files, social sites	images that link to other websites (shopping function)	not described	video preview, timeline-based pattern, input forms, tools, widgets, media, player preview
wireWAX [Wir12]	"taggable video tool"	linear video	jumps to markers for hotspots on timeline	no/not described	image, text, video, links to external pages	different shapes, can move with objects in video, show additional information	not described	video preview, WYSIWYG-tagging, tag editor-form

5minMedia VIDEO EVERYWHERE [5mi14]	instructional videos, maximum length: 5 minutes	one linear video	by the definition of scene entry-points	list of scenes	text, links and images as overlays or in side-area, add-ons, scenes	combination of image and text, fixed position, link to other websites	not described	severals editors
ConnectME [NBB13]	"annotation tool"	linear video	no/not described	no/not described	load additional information screens on a button click (image, text, video, links to external pages)	no/not described	annotations are RDF, LOD identifiers for concepts	video preview, timeline view
Popcorn Maker [Moz13a]	"lets users link social media, news feeds, data visualizations and other content directly to moving images. The result is a new form of multimedia storytelling [...] interactive, social, and unique each time."	video	no	no	text, images, Google Maps, Twitter, social websites	combination of image and text, fixed position, link to other websites	HTML5 (+JavaScript)	parallel timelines for annotations, annotation editing area, video preview

Table C.2.: Authoring tools for interactive videos.

Authoring Tools for Hypervideos

Source/cite	*Scope*	*Main medium/ video scenes*	*Non-linear structure (defined by author)*	*Alternative playback paths (jumps)*	*Influence on order of scenes (before playback)*	*Choice elements*	*Additional information*	*Hotspots*	*Language*	*Patterns/editors*
HyperVideo Linking Generator (HVLG) [Hun97]	"hypervideo system generator for automatic implementation of various hypervideo systems"	video scenes in graph structure	graph defined by hyperlinks	jumps triggered by hotspots, jumps to frame numbers	no/not described	rectangled hotspots	video (.avi), audio files (.mid), images (.bmp), sound (.wav)	jumps between scenes and to audio files, sound, and images	"self defined specification language ("Hyperlink data structure", (GVHS))"	video preview, tabular view for links, hotspot-list
HyperProp [SRMS00]	"represent context relations, synchronization relations, derivation relations and task relations in hypermedia systems. It discusses temporal and spatial synchronization among multimedia objects"	media files	defined by hyperlinks	no/not described	no/not described	yes?	text, graphic, audio files, video, etc.	no/not described	NCM/NCL	structural view, temporal view, spatial view
Hyper-Hitchcock [SGW03b; SGW03a; SGW05; SGW08]	"Detail-on-demand video is a form of hypervideo that supports one hyperlink at a time for navigating between video sequences" (detail-on-demand video)	video	defined by several types of links	defined by links and user behavior	no/not described	keyframes of linked videos	videos	no/not described	not described	timeline, clip selection panel, tree view, workspace

Chang et al. [Cha+04]	"Video objects can be described by semantic annotation and multistory movies can be produced.", "User defined video object annotation", "Multistory video viewing"	one linear video	based on annotations	"choose an annotated region in a segment to he a 'branch point'"	no/not described	hotspots	"multimedia descriptions", "additional data can be a text, a video clip, a URL link, or a still image"	used for jumps in the video (to other scenes)	not described	graph view, video preview, overview for defined video pieces
Finke and Balfanz [FB04]	"reference architecture supporting hypervideo content for ITV and the internet domain"	list of video scenes	"The navigation engine provides functionalities for the support of orientation within the presented content."	jumps between scenes (previous scene, next scene), jumps on timeline	no/not described	not described	"any form of information media", HTML	rectangled hotspots which track objects	hypervideo metadata model, data model, data repository	not described
Zhou et al. [ZGJ05]	"[...] automatic generation of additional information and the integration of the additional information to its corresponding selectable video object" (detail-on-demand video)	video	no/not described	yes	no/not described	none in videos, but links in annotations	video frame images and html files	no/not described	MPEG-7, SMIL	converter view with two tree structures
Advene [AP05; AP07; APS12]	"active reading applied to audiovisual material can be hypervideos, that we define as views on audiovisual documents associated with an annotation structure"	one audiovisual document	no	jumps in the audiovisual document defined by annotation layer	depending on annotations	no?	annotations rendered to different views	no/not described	own model, no standard	stream-time based view, view for note taking, tree view, parallel timelines, description area, video area

Hsu et al. [Hsu+05]	hyper-interactive video browsing by a remote controller and hand gestures	video scenes in graph structure	graph	no/not described	no/not described	"hyperlink in a specified temporal-spatial domain"	text descriptions, existing image files, web page files or URLs	no/not described	not described	video preview, annotation area, graph view
HyPE and Jeherazade [HH06]	"there seems to be a lack of using narrative intelligence in hypervideo. This paper shows how both fields of work could benefit from each other."	linear video	no/not described	jumps triggered by hotspots	no/not described	hotspots	video or audio player, a text or an image window	for jumps in the video and to display additional information	XML file, no standard	video view, list with hotspots (polygon)
Klynt [Hon13]	"editing & publishing application dedicated to visual storytellers. It was designed as an affordable and creative solution to explore new narrative formats on the internet."	video scenes in graph structure	visual storyboard to create a graph	links between sequences	no/not described	buttons on the video, several menus	text, graphic, audio files, video, hyperlinks	no/not described	not described	graph view, WYSIWYG editor, timeline view
LinkedTV/Video-HypE [Lin13] [RGT13]	"LinkedTV is an integrated and practical approach towards experiencing Networked Media in the Future Internet"	video scenes	list of video scenes	hyperlinks between video segments	no/not described	no/not described	multimedia content (on second screen)	no/not described	LinkedTV ontology, Media Fragments URI, RDF, and NER	chapter editor, timeline view, link editor

Table C.3.: Authoring tools for hypervideos/hypermedia.

Authoring Tools for Clickable Videos

Source/cite	*Scope*	*Main medium/video scenes*	*Alternative playback paths (jumps)*	*Additional information*	*Hotspots*	*Language*	*Patterns/editors*
Overlay.TV [Ove10]	"place an interactive layer of clickable hotspots on top of video allowing your customers to shop directly from the video"	linear video	no/not described	image, text, links, shopping-cart	image of object in video, shows additional information for an object in the video, shopping-option	not described	WYSIWYG-editor, hotspot-editing function, input fields for annotation-content, preview, annotation repository
Viddix Beta [VID10]	"[...] connect all kinds of web-content to your videos. This way you can really interact with your audience and deliver your messages more effectively"	linear video	no/not described	text, link, image, rss feed, poll, html-page (may be clickable and linked with web page)	hotspots at fixed position in the video (rectangles), show additional information or link to web page	not described	video preview and timeline, cuepoint-editor with configuration for annotation
VideoClix [Vid12]	"[...] allows your viewers to immerse themselves in your content. Every object is clickable enabling your audience to learn, shop, play and vote while they watch video"	linear video	no/not described	image, text, voting, link to website (online shop)	any object in the video (automatically detected and tracked), shows additional information	not described	video preview, overview over detected objects, forms to describe objects (annotations)
Klickable [Kli13]	"Klickable videos create a more engaged user and a comprehensive viewing experience"	video	no	text, images, links to external pages, shopping cart	rectangles which invoke additional information	not described	WYSIWYG-editor, hotspot-editing function, timeline, controls, input fields for annotation-content

Table C.4.: Authoring tools for clickable videos.

Authoring Tools for Multimedia Presentations

Source/cite	*Scope*	*Main medium/video scenes and additional information*	*Alternative playback paths (jumps)*	*Non-linearity and interaction*	*Language*	*Patterns/editors*
Harmony [Fuj+91]	synchronization and timing in multimedia presentations, link semantics	"text, music, graphics, motion video, and computer animation"	not described	non-linear structure (defined by author): yes, choice elements: not described	own model, no standard	not described/not visible in screenshot
TYRO [Mac91]	"capturing multimedia design knowledge and reusing it to make automatically, generated presentations"	video, audio files, images, text	no/not described	no/not described	not described	temporal editor, rule editor, condition editor, image browser, spatial editor, narration-browser, midi-score-browser, script-editor
The Synchronization Editor [BHL92]	synchronization of multimedia objects	sequential parallel media	no/not described	no/not described	"synchronization model based on synchronization at reference points [...] stored in text form following a syntax defined in a context free grammar [...]. This allows usage of the synchronization specification by MODE components independent of their implementation language and environment."	presentation view, time view, layout view
CMIFed [Ros+93]	"Unlike systems that use a timeline or scripting metaphor to control the presentation, in CMIFed the user manipulates a collection of events and timing constraints among those events"	mixture of text, images, audio files, and video (and possibly other media)	defined by hyperlinks	some form of user input	CMIF model for hypermedia documents	hierarchy view, channel view

Eventor [Eun+94]	"focus on describing the temporal and spatial synchronizations and user interactions"	still image, motion video, text, audio files	no/not described	"buttons, menus, and others"	calculus of communicating systems (CCS) [Mil89]	Temporal Synchronizer, Spatial Synchronizer, User Interaction Builder
Delaunay MM [CL97]	"querying and presenting multimedia information in distributed databases"	media elements (i.e. text, image, video, and audio objects)	no/not described	influence on order of scenes (before playback): document generation module	no/not described	not described/not visible in screenshot
Madeus [Jou+98]	"[...] efficient support for the specification of temporal scenarios and this in an architecture that allows the integration of both authoring and presentation phases of multimedia documents.", "constraint-based",	"Mpeg audio and video, different image formats and formated text"	"Temporal navigation [...]: Context dependent navigation [...] and Context independent navigation"	not described	Madeus language (XML)	timelines, graphs, multiple views (no screenshots available)
IMMPS [SD97]	"[...] uses artificial intelligence to specify knowledge inheritance relations between presentation windows. An objectoriented [sic] multimedia database organizes resources and presentations, and a database browser facilitates object reuse."	audio files, video, text, image, "knowledge"	yes? depending on answers	influence on order of scenes (before playback): answer to questions decide on content; choice elements: various buttons depending on scenario	self defined specification language	presentation knowledge inheritance window, presentation message passing window, multimedia resource browser
MPRES Author [WRR97]	"multimedia presentation system that allows a user to compose and render a presentation consisting of objects referenced by their URLs (Uniform Resource Locators)"	"audio files, image, Hypertext Markup Language (HTML) document, plaintext or animation", "titles and background"	no/not described	no/not described	self-defined	various input masks
MediaTouch [Ech+98]	"a visual-based authoring tool [...]. It's based on the native approach, which lets the author operate at the level of MHEG-5 objects."	scenes, media elements in a tree structure	links between elements/scenes	hotspots, hyperlinks	MHEG-5	Hierarchy Editor, Properties Editor, Layout Editor, Links Editor,

Shih et al. [Shi98; Shi+98b; Shi+98a]	"dynamic multimedia presentation can learn from an audience and act according to the audience's individual behavior", "collection of some Petri nets, which are simulated in our Petri net engine"	sound, video, animation, picture, text	no/not described	de-	no/not described	no standard (rules defined in parameterized temporal interval relations)	multimedia resource browser, temporal specification editor, presentation story board, spatial specification editor
GRiNS [Bul+98]	"[...] allows the original media assets to be allocated to screen locations [...], and have their presentations synchronized [...] presents a hierarchy of the node structure of the multimedia document to promote the re-use of its components [...] presents the hyperlink structure within the document and pointing to the outside web"	media files	no/not described	de-	no/not described	SMIL	end user view, hierarchical structure view, timeline view (channel view), hyperlink view
Villard [Vil01]	"method for authoring generic and adaptable multimedia presentations"	"media objects (text, audio files, 3D animation, etc.)"	no/not described	de-	no/not described	"Madeus model"	expression view, execution view
Deng and Shih [Den+02b]	how to present different multimedia objects on a web-based presentation system	linear video and slides	no/not described	de-	no/not described	extended timed Petri Net	various input masks (no pattern)
Gaggi and Celentano [GC02]	"set up and test a complex multimedia presentation by defining the synchronization relationships among media"	continuous media, like video or audio clips, and non-continuous media like images or text pages annotated with various media, parallel and sequential execution	no/not described	de-	non-linear structure (defined by author): partially, provided by hyperlink activation	XML file, no standard	timeline represented as a tree, graph view, layout view
SMILAuthor/ SMILAuthor2 [YY03; YCW04a; YCW04b; YCW08]	"parsing process extracts and converts the temporal relationship of the input script to Real-Time Synchronization Model (RTSM), and the playback duration of each object in the script is then computed by traversing the RTSM"	media files	no/not described	de-	no/not described	SMIL	visual layout window, timeline window, filter window, attribute window, preview window

Sung and Lee [SL05]	"Java-based collaborative authoring system for multimedia presentation"	various media (video, animation, images, text, sound)	no/not described	no/not described	SMIL	Media Object Manager, Collaboration Manager, 3D Spatio-Temporal editor, Temporal Relation Network (TRN) editor, timeline editor, tag editor, attribute editor, text editor
Java-Assisted SMIL (JAS) [DTL06]	"advanced education supporting tool that ensures the maximum flexibility and deliverability for building multimedia presentation in an e-learning system"	one linear video with images, text, links to web resources	no/not described	no/not described	SMIL	timeline, preview for video and images/text
SIMPLE [Mur+06]	"reference information of many types, at varying granularity, without replicating the referenced information. [...] compose synchronized multimedia presentations"	"multimedia information"	no/not described	selection of contents	"uses XML for storing presentation data"	no description available
Jokela et al. [JLK08]	"makes it possible to author sophisticated multimedia presentations that integrate several different media types on mobile devices"	images, stickers (small icons), texts and text bubbles, audio files, video (future work); parallel playback of static media and audio files	no/not described	non-linear structure (defined by author): linear structure	SMIL	timeline, several lists, preview
MEMORY [KHM08]	"integrated approach for adaptive multimedia presentations enabling universal access for situational learning"	various media (audio file, video file, XML file, PDF file, DOC file)	"Navigation possibilities for jumping to different media documents or fragments presented in a hit list"	no/not described	LOM	not described/not visible in screenshot

Cutts et al. [Cut+09]	"use of a video segmentation process that provides contextual supplementary updates produced by users. Supplements consisting of tailored segments are dynamically inserted into previously stored material in response to questions from users"	multimedia documents, supporting text (with links), frequently asked question	table of contents, search, marker on timeline	no/not described	XML files, no standard	not described/not visible in screenshot
LECTURNITY 4 [imc10]	"creen recordings for software training and e-learning content for company training, to e-lectures for teaching and sales training productions"	Powerpoint-presentation, audio files, video, images	"user navigation via buttons, transparent interaction and rollover areas", "navigation possibilities within the document: directory, thumbnails, timeline, title, keyword and full text searches"	choice elements: hotspots for navigation within the presentation	not described	parallel timelines, preview-area, toolbar
SMIL Builder [BB11]	"[...] temporal SMIL editor with incremental verification capabilities, based on a formal Petri Net-based model. [...] build his document step by step, while insuring at every stage the validity of the current state of the document"	media files	no/not described	no/not described	SMIL	hierarchical view, textual view, attributes view, temporal view, message zone
Matchware Mediator 9 [Mat12]	"create interactive CD-ROM presentations, dynamic HTML pages and Flash projects [...] icon-based editing, [...] without requiring any coding or scripting"	media files	no/not described	non-linear structure (defined by author): yes; choice elements: links, menus; hotspots: "make them interactive by hyperlinking to a website or going to another section for further explanation"	HTML (+JavaScript)	WYSIWYG-editor, hotspot-editing function, parallel timelines, toolbar, input fields for annotation-content, event-editor, animation-editor

NextSlidePlease [Spi+12]	"authoring and delivering agile multimedia presentations"	presentation slides	jumps between related/linked slides	graph structure, next slide definition	not described	Overview Inset, Time Cost and Priority Controls, Presentation Graph Editor Plane, Zoom Slider, Graph View
ZEEGA [Zee13]	"With Zeega, you can use any media in the cloud, transform the entire screen into your playground, and share your interactive creations with the world."	mixture of text, videos, animated images, and audio files, (and possibly other media)	no/not described	user clicks next button to see next screen	not described	WYSIWYG view for one screen, list of screens

Table C.5.: Authoring tools for multimedia presentations.

D. Player

Player for Non-linear Videos

Source/cite	*Kind of player*	*Scope*	*Main medium/video scenes*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Player controls*	*Additional information*	*Language*
XIMPEL [Bhi+10]	web player	"create interactive media applications" , "create their own storylines"	video scenes	non-linear structure, graph	different hotspots at the same time, invoke next scene, polls, questions	pause/play, forward, rewind, full screen	text and image, display of score; overlays over video	self-defined XML-format
Riva player [mem13]	web player, stand-alone player	"splitting the videos into small information units and by the use of annotations for each unit you get a video-based database"	videos, cut into scenes	non-linear structure	hotspots which lead to another scene	play/pause, forward, rewind, timeline, volume control	graphics and buttons in the video; one video area, no areas for additional information	own XML-format
YouTube Video Annotations [You13]	web player	"add interactive commentary to your videos": "add background information about the video", "create stories with multiple possibilities", "link to related YouTube videos, channels, or search results from within a video"	linear YouTube-videos	graph (not visible, created by links)	different types of hotspots link to another YouTube video or homepage	play/pause, volume control, timeline, hide annotations, change quality, watch later, size	"speech bubble, note, title (text), spotlight (hotspot), label; overlays over video"	not described

Table D.1.: Player for non-linear videos.

Player for Interactive Videos

Source/cite	*Kind of player*	*Scope*	*Main medium/video scenes*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Player controls*	*Additional information*	*Language*
HyStream System [Bea+02]	"HyStream Browser"	link creation for continous media, approach enriches hypermedia content with additional metadata, only "Seminar Application" in paper	one linear video	linear video with jumps in annotations (presentation slides)	no/not described	video player with play and pause; navigation forward and rewind for slides	small video area, larger presentation area, presentations with slides and labeled links (with start- and end-point)	RDF
LazyMedia [HL06]	web player	export result of creation process as a web page	scenes combined to one linear video	alternative playback paths (jumps) based on video chapters	video chapters	embedded player with standard controls, clickable video chapter images	text, photos, audio files	LMPF file
Chang et al. [CHS07; CHC08]	stand-alone player (Video SCORM Player)	"The gaming platform is an augmented video player with the interaction functionalities. The users can [...] make interactions with the interactive objects."	video files which are divided into scenario components	jumps: "buttons [...] to switch to other video segments"	buttons/images that appear in the video	play, stop, previous, next, resume	images, links to websites	not described
EmoPlayer [Che+08]	Stand alone player (implemented with VC++ and DirectShow)	"select a character in a video clip and view the distribution of his/her emotions along the video timeline through a colour based interface"	linear video	no/not described	"combo box which can be used to switch between different characters"	play/pause, stop, combo box, time display, process bar, legend for emotions	affective annotations	XML file (no further description)
HyLive [HKH08]	player for TV or standalone player with Flash	"a web-based client player for interactive live television with hypervideo structures"	linear video	no/not described	hotspots: rectangled areas which open additional information	not described	elements for interactions like voting and hypervideo links which refer to additional information about the content	not described

Composer [LGaMDRCGS08]	TV middleware	"when integrated to a TV middleware, [...] extend the traditional television-watching experience [...]. New contents can be added, new links etc., without changing the original content received."	video/media	no detailed description of player available				Nested Context Language 3.0
Chen et al. [Che+09a]	Player with several different interfaces	"combine the video-based course materials and game elements with an integrated learning platform"	sequential linear videos	not described	interactive elements that have to be used to access the next scene, no hotspots	several controls and buttons depending on the task to solve	games, images, text	SCORM
SeViAnno [Cao+10]	mixture of player and annotation tool, browser application with flex plugin	"complete annotation functionality which includes creation, display and editing"	one linear video	alternative playback paths (jumps) implemented as navigation by clickable annotations	no/not described	play, sound volume, full screen, timeline	"All annotations are shown in a list sortable by type, time point or alphabetically. Place annotations are shown on the lower right side in the integrated Google Map. [...] The description and keywords are displayed on the upper right side of the application [...]"	MPEG-7
Räck et al. [RSA10]	web based stand-alone player	"The media player highlights all objects, which have been previously identified and which are described in the corresponding metadata.", "The system provides multiple interaction layers, [...] he can navigate throw [sic] the media."	video	no/not described	hotspots used to show additional information with an object in the video	standard controls, timeline, buttons for additional information	image, text; displayed as an overlay over the main video	Object Definition Language (ODL)
ADIVI Production Kit [Inn11]	web player	"This type of multimedia presentation merges the advantages of website and video in one single media."	video	no, only one linear video	rectangles or circles which invoke additional information	timeline, full screen	yes, "different multimedia information like additional videos, documents, pictures etc."	not described

Quicktvpro [Bel12]	web player, player-plugin	"interactive video effects"	linear video	jumps in the video to beginning of chapters	clickable areas, jumps to other scenes, hotspots in form of clickable images or texts which invoke other sides or save results	play/pause, timeline, volume control	voting and polling, links to sales webpages, images, text, shapes, SWF files, social sites; displayed as overlays over main video area	not described
wireWAX [Wir12]	web player, player-plugin	"taggable video tool"	linear video	jumps to markers for hotspots on timeline	hotspots in different shapes, can move with objects in video, show additional information	depending on player	image, text, video, links to external pages; overlays over video area	not described
5minMedia VIDEO EVERYHWERE [5mi14]	web player	instructional videos, maximum length: 5 minutes	video	jumps by the definition of scene entry-points	list of scenes, hotspots as a combination of image and text, fixed position, link to other websites	restart, pause/play, timeline, volume control, video resolution, share menu, full screen, related, tools-menu, smarts menu, search	yes, text, links and images as overlays or in side-area; add-ons, scenes, links	not described
Popcorn Maker [Moz13a]	web player	"lets users link social media, news feeds, data visualizations and other content directly to moving images."	video	no, only one linear video	hotspots as a combination of image and text, fixed position, link to other websites	play, timeline, share, volume control, remix, full screen	text, images, googlemaps, twitter, social websites; freely positioned	HTML5 (+JavaScript)

Table D.2.: Player for interactive videos.

Player for Hypervideos

Source/cite	*Kind of player*	*Scope*	*Main medium/video scenes*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Player controls*	*Additional information*	*Language*
HyperVideo Linking Generator (HVLG) [Hun97]	stand-alone player	"When the video in a hypervideo system is played back, the viewer can trigger a hyperlink and jump from frame to frame."	video scenes in graph structure	non-linear structure defined by hyperlinks, jumps to frame numbers	rectangled hotspots (fixed position for a defined frame range)	standard controls	image, sound, audio files, video displayed in main area	self defined specification language ("Hyperlink data structure", (GVHS))
HyperProp [SRMS00]	??	"The spatio-temporal formatter, or simply formatter, is responsible for controlling the document presentation based on its specification and on the platform (or environment) description"	media files	no detailed description of player available				NCM/NCL
Hyper-Hitchcock player [SGW03b; SGW03a; SGW05; SGW08]	stand-alone player	"The Hyper-Hitchcock player was iteratively developed over several user studies. [...] The resulting design included keyframes and link labels to help viewers rapidly navigate and orient themselves."	video	non-linear structure defined by several types of links, alternative playback paths defined by links	keyframes of linked videos, "All keyframes are clickable, thus enabling the user to return several link levels at once.", no hotspots	play, stop, navigation buttons, timeline, keyframes	videos, displayed in area of the "main video"	not described

Chang et al. [Cha+04]	stand-alone player	"Multistory video viewing", "A video player is also developed for the video viewer to view the annotated film efficiently."	one linear video	non-linear structure based on annotations, "choose an annotated region in a segment to be a 'branch point'"	hotspots used for jumps in the video (to other scenes)	cannot be determined	yes, "multimedia descriptions", "additional data can be a text, a video clip, a URL link, or a still image"; two-part view with annotations on right side	not described
Finke and Balfanz [FB04]	Web player	"client side consists of an annotation engine and the presenter engine. The current system requirements for an executable hypervideo environment on the client side are an ordinary Web browser, a QuickTime plug-in that enables the video presentation of the video streaming server, and a java [sic] virtual machine, basically for the presentation of the video and annotations [...]"	video scenes in linear order	navigation in video via timeline, jumps to next and previous scene	hotspots to invoke additional information	play/pause, jump forward/ rewind, navigation view with hyperlinks	rectangled hotspots as overlay over video, text, images, etc. in HTML pages/subpages	not described, HTML for annotations
Advene [AP05]	stand-alone player, interactive homepage	static view: "definition of a hypertext document, whose temporality is imposed by the user visualising [sic] it", dynamic view: "the temporality of the resulting document is mostly imposed by the audiovisual document. [...] interaction opportunities [...]. STBV can be seen as a video augmented with additional capabilities."	one audio-visual document	navigation in video via timeline, URLs	no/not described	standard controls, hyperlinks, URL stack, navigation links, position indicator	shown around video, overlay over video, mainly text-based	own model, no standard
Hsu et al. [Hsu+05]	stand-alone player	hyper-interactive video browsing by a remote controller and hand gestures	video scenes in graph structure	graph	"hyperlink in a specified temporal-spatial domain"	gesture controls	text descriptions, existing image files, webpage files or URLs on the Internet	not described

HyPE stand alone player [HH06]	stand-alone player (HyPE stand alone player)	"The hypervideo player loads and starts the basic video and the meta data information."	linear video	jumps triggered by hotspots	hotspots for jumps in the video and to display additional information	none	two-part window: video or audio player, a text or an image window	XML file, no standard
Klynt [Hon13]	Web player	for visual storytellers, to "explore new narrative formats on the internet"	mixture of video centered multimedia presentation and hypervideo, depends on the realized project	navigation in video via buttons, jumps to other scenes or short presentations, navigation with Google map	buttons on video canvas	play/pause, timeline, full screen, volume control, social media, (menus)	overlay over video, text, images, web elements	not described
LinkedTV [RGT13]	player with second screen	browsing and navigation with second screen	video on first screen, navigation and additional information provided by links on second screen	navigation in video via timeline, jumps to other scenes on second screen	no/not described	controls on second screen, external control interfaces	shown on second screen, mainly text-based or websites	LinkedTV ontology, Media Fragments URI, "RDF (a semantic data model) and NER (Named Entity Recognition)"

Table D.3.: Player for hypervideos/hypermedia.

Player for Clickable Videos

Source/cite	*Kind of player*	*Scope*	*Main medium/video scenes*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Player controls*	*Additional information*	*Language*
Overlay.TV [Ove10]	web-player	"place an interactive layer of clickable hotspots on top of video allowing your customers to shop directly from the video"	linear video	no/not described	image hotspots of object in video, shows additional information for an object in the video, shopping-option	play/pause, timeline, volume control, info, share	image, text, links, shopping-cart, "collaborative text annotations in "experience pages" positioned next to video"	not described
Viddix Beta [VID10]	web-player	"[...] connect all kinds of webcontent to your videos. This way you can really interact with your audience and deliver your messages more effectively"	linear video	no/not described	hotspots at fixed position in the video (rectangles), show additional information or link to web page	play/pause, timeline, volume control, full screen	text, link, image, rss feed, poll, html-page (may be clickable and linked with web page); annotations as overlay over the video or two-part view with video on left side and annotation on right side	not described
ConciseClick [Cle12]	web-player	"enables clickable videos of your content to enable faster purchasing decisions"	linear video	no/not described	click on object shows additional information for the object in the video, shopping-option	play/pause, timeline, volume control, share (on Facebook)	image, text, links to external page for shopping	not described
VideoClix [Vid12]	web-player	"[...] allows your viewers to immerse themselves in your content. Every object is clickable enabling your audience to learn, shop, play and vote while they watch video"	linear video	no/not described	hotspots: any object in the video, shows additional information	play/pause, timeline, volume control, full screen, list of objects, share, settings, recommended videos	image, text, voting, link to website (online shop); shown as overlays over main video; no additional areas for forms invoked by buttons, shown as overlay	not described

Klickable [Kli13]	web player	online shopping and video enrichment	linear video	no	rectangled hotspots which move with the object, show additional information; clicked objects are collected in a list as links to websites	play/pause, timeline, volume control	text, images, links to external pages, shopping cart; displayed at the bottom of the video area	not described

Table D.4.: Player for clickable videos.

Player for Multimedia Presentation

Source/cite	*Kind of player*	*Scope*	*Main medium/video scenes*	*Navigation, structure, influence on order of scenes*	*Interaction with video (choice elements, hotspots)*	*Player controls*	*Additional information*	*Language*
TYRO [Mac91]	preview in authoring tool	"The Spatial Editor can be used to do a trial playback of a sequence to check timing."	video, audio files, images	no/not described	no/not described	no detailed description of player available	video, audio files, images, text; freely positioned	not described
Blakowski et al. [BHL92]	stand-alone player	"[...] perform the synchronized presentation [...] according to the introduced synchronization model. This comprises the intra-object and the inter-object synchronization.", "supports joining of different basic objects."	sequential parallel media	no/not described	no/not described	standard controls (restart, quit, forward, rewind, pause, play), timeline	media files	"[...] syntax defined in a context free grammar (Synchronization Description Language). [...] usage of the synchronization specification by MODE components"

CMIFed [Ros+93]	player preview in authoring tool	"the player, shows the effect of mapping the abstract document to a particular platform."	mixture of text, images, audio files, and video (and possibly other media)	non-linear structure defined by hyperlinks	no detailed description available/not described	standard controls, buttons for options and channels	"mixture of text, images, audio, and video [...] displayed in a control panel and additional windows for screen-oriented channels", "normally shows one window per screen-oriented channel", size and position initially set by author, can be changed and saved by viewer	CMIF model for hypermedia documents
CAI application (Eventor) [Eun+94]	computer aided instruction (CAI) application	"illustrate that our specification mechanism is well-suited for handling the interactivity of multimedia applications"	still image, motion video, text, audio files	no/not described	no/not described	not described	still image, motion video, text, audio files; positioned as (moving) overlay	CCS
IMMPS [SD97]	description not apparent from paper	"An object oriented multimedia database organizes resources and presentations, and a database browser facilitates object reuse."	audio files, video, text, image, "knowledge"	jumps and content depending on answers of users to questions	various buttons depending on scenario	description not apparent from paper	audio files, video, text, image, "knowledge"; displayed in several windows with buttons and media elements	self defined specification language
Madeus [Jou+98]	"Madeus presentation engine"	"One of the main goals of the presentation engine of Madeus is to dynamically adapt to the current presentation conditions."	media objects	no detailed description of player available				XML representation of Madeus object model
MPRES Viewer [WRR97]	web interface (developed for Netscape WWW browser)	"WWW interface which allows a user to access various options, such as, 'play' a particular presentation, 'browse' through a list of available presentations stored in the presentation database and invoke the authoring tool to edit/compose presentations."	"multimedia objects of types such as audio, image, [...] (HTML) document, plaintext or animation", "titles and background"	no/not described	no/not described	play, browse, edit/compose presentation	"multimedia objects of types such as audio, image, Hypertext Markup Language (HTML) document, plaintext or animation", audio, image, text and HTML, animation, titles and background; spatial layout not apparent from paper	selfdefined

GRiNS [Bul+98]	player preview in authoring tool	"The end user view provides a WYSIWYG view of the presentation under development, as well as a mechanism to interactively lay out the spatial position of the layout channels associated with the presentation."	media files	no/not described	no/not described	WYSIWYG view of the presentation under development, no standalone player	media files	SMIL
Deng and Shih [Den+02b]	web player	how to present different multimedia objects on a web-based presentation system	linear video	no/not described	no/not described	no detailed description of player available	no detailed description of player available	extended timed Petri Net
Gaggi and Celentano [GC02]	execution simulator	"An execution simulator allows the author to check the temporal behavior of the presentation. The simulator places the media placeholders in the appropriate channels they would use in the real execution."	continuous media (video, audio files), non-continuous media (images, text pages)	non-linear structure partially provided by hyperlink activation	no/not described	simulator with start, pause, end, stop, reset, import, close	simulator only shows placeholders for media elements	XML file, no standard
AMBULANT SMIL (2.0, 2.1, 3.0) player [Bul+04; CWI10]	stand-alone player, browser plugin	"reconfigurable SMIL engine that can be customized for use as an experimental media player core"	media files	depends on SMIL file	depends on SMIL file	open file, play, pause, stop, view source	media files; positioned as defined in SMIL file	SMIL
SIMPLE [Mur+06]	"play" interface in authoring tool	"reference information of many types, at varying granularity, without replicating the referenced information.", play "synchronized multimedia presentations"	multimedia information	no/not described	no/not described	controls for single media	video, audio files, images, text; freely positioned	self-defined XML-format

Jokela et al. [JLK08]	combination of mobile player and authoring tool	"The Play Presentation View enables the user to view the selected presentation. The presentation player displays over the full screen when playing a presentation."	"images, stickers (small icons), texts and text bubbles", audio files, video (future work)	linear structure	no/not described	de-	pause playback, restart playback from beginning	parallel playback of static media and audio files, two-part view	SMIL
MEMORY [KHM08]	web-based presentation tool	"reacts during runtime to events created by adaptation triggers (interaction, context change, time, content presentation). Media fragments to be presented are dynamically determined during execution of the presentation process, and adapted to user interaction and context."	various media	"Navigation possibilities for jumping to different media documents or fragments presented in a hit list"	no/not described	de-	video: play, pause, stop; additional information: forward, rewind, other buttons	audio file, video file, XML file, PDF file, DOC file; video area and information area	LOM
Cutts et al. [Cut+09]	standalone player with editing function	"The student requires not only access to the audio/video segments but also a measure of control over their delivery. Being able to select and re-run segments [...] intuitive navigation system [...] a table of contents with associated support text."	multimedia documents	alternative playback paths by table of contents, search, marker on timeline	no/not described	de-	forward, rewind, play, timeline	supporting text (with links), frequently asked questions, table of contents; fixed areas	XML files, no standard

LECTURNITY 4 Player [imc10]	stand-alone player	"With our Player, you can use the unique advantages of the lpd file format, such as the search function or the full scalability of slides."	powerpoint-presentation	navigation with "buttons, transparent interaction and rollover areas", "directory, thumbnails, timeline, title, keyword and full text searches"	hotspots for navigation within the presentation	standard controls for video, buttons/hotspots in presentation area	audio files, video, images; view divided into several parts	not described
RealPlayer 16 (SMIL 2.1) [Rea12]	stand-alone player	"Play music and video files, and display photos"	media files	no detailed description of the playback of SMIL files in the player available				SMIL
Chrooma+ [Oeh+13]	player for web browser	"time-oriented composition of media streams with HTML components or widgets" "highly adjustable and provides support for extensions"	video	no/not described	no/not described	basic controls (play, timeline, volume control, full screen)	connectors for different services (Google Maps, Twitter, Wikipedia, etc.)	HTML5, WebVTT

Table D.5.: Player for multimedia presentations.

E. Sequence Chart of Interactions in the Framework

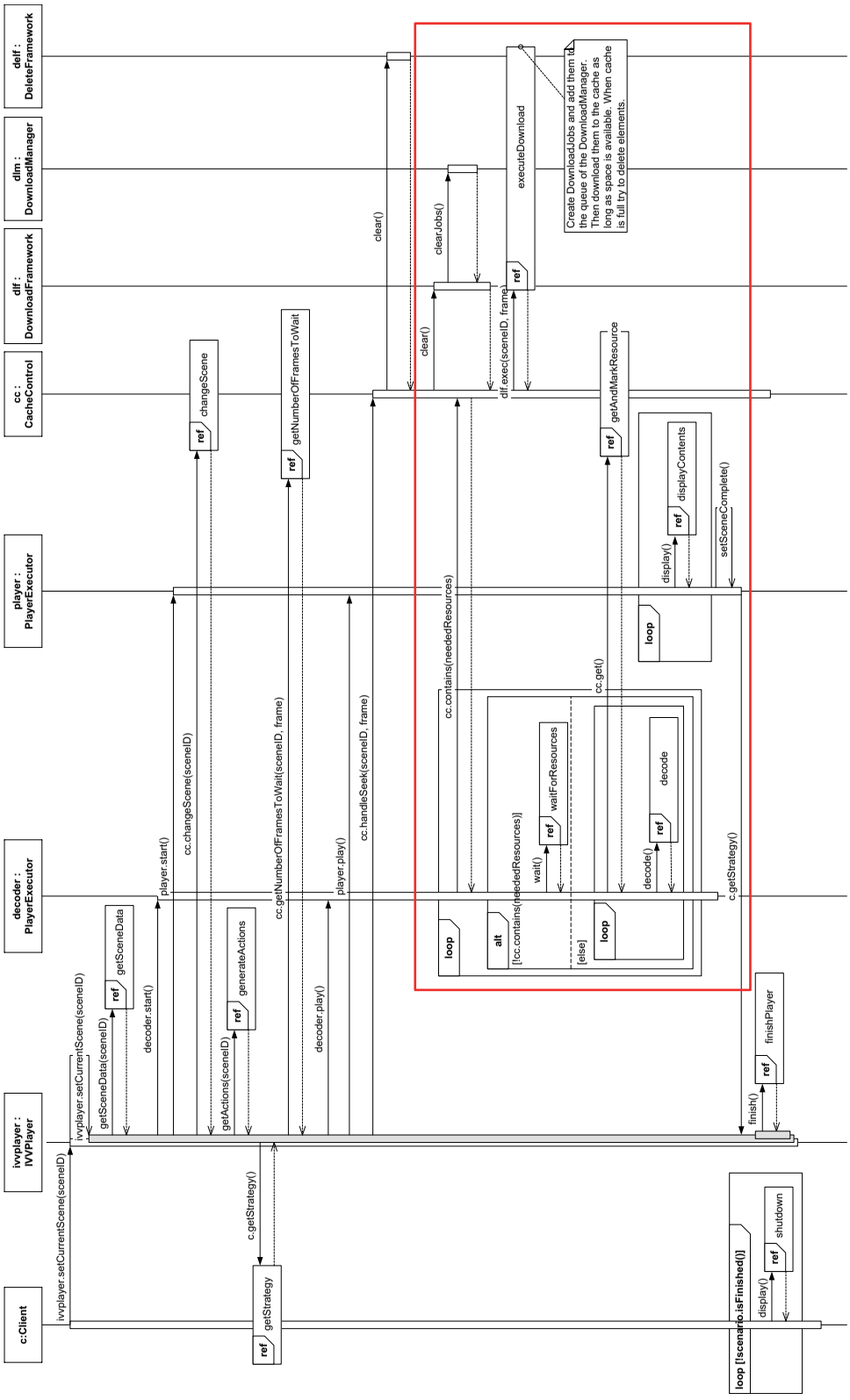


Figure E.1.: Simplified UML sequence diagram illustrating the interactions between the most important classes of the simulation framework.

F. Settings of the User Generated Scenarios

The sizes of the frames and annotations in the user generated scenarios are calculated from the resolution and the color depth selected in the second part of the graph creation task. Resolutions and color depths were derived from real world examples. Therefor, the resolution and the color depth are multiplied for the frames (see Equation F.2). The sizes of the annotations are the result from the frame sizes multiplied with a fixed factor. The used factors are given in Equation F.1 and derived from real world values. The used frame rate is 25 fps as defined in Equation F.3.

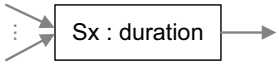
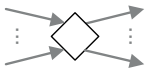
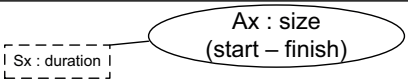
$$s(a_x) := \begin{cases} Resolution \cdot Color\ depth \cdot 2, & \text{small Annotation} \\ Resolution \cdot Color\ depth \cdot 10, & \text{medium Annotation} \\ Resolution \cdot Color\ depth \cdot 25, & \text{large Annotation} \end{cases} \tag{F.1}$$

$$s(f_x) := Resolution \cdot Color\ depth \tag{F.2}$$

$$c_r := 25\ fps \tag{F.3}$$

Graph Creation Task

Description – Task 1:
Paint a valid scene graph from the elements given below. The start and the end element can be used once. Use all other elements as often as you want (each at least three times) without violating the rules for the edges and links.

Element	Allowed edges/links
Start 	1 outgoing edge
Scene 	n ingoing edges 1 outgoing edge
Fork 	1-n ingoing edges 2-n outgoing edges
End 	1-n ingoing edges
Annotation 	1-n links to (different) scenes

Description – Task 2:
Select 3 not necessarily disjoint combinations of resolution, color depth, bandwidth, and cache.

resolution:	*color depth:*	*bandwidth:*	*cache:*
		5,76 Mbit/s	64 MB
2560 x 1440		10 Mbit/s	128 MB
1920 x 1200			
1920 x 1080	8 bit	16 Mbit/s	256 MB
1680 x 1050			
1600 x 900	16 bit	25 Mbit/s	512 MB
1440 x 900			
1366 x 768	24 bit	32 Mbit/s	768 MB
1280 x 1024			
1280 x 800	32 bit	50 Mbit/s	1024 MB
1024 x 768			
800 x 600		100 Mbit/s	2048 MB
		1000 Mbit/s	4096 MB

Description – Task 3:
Give your scene graph to your neighbour. Then, add probabilities to all fork alternatives which sum up to 100% for each single fork element.

Description – Task 4:
Give the scene graph to your neighbour. Then, use five different colors to paint five paths into the scene graph. Start each path at the start element and finish each path at the end element.

Figure F.1.: Task description for the graph creation task.

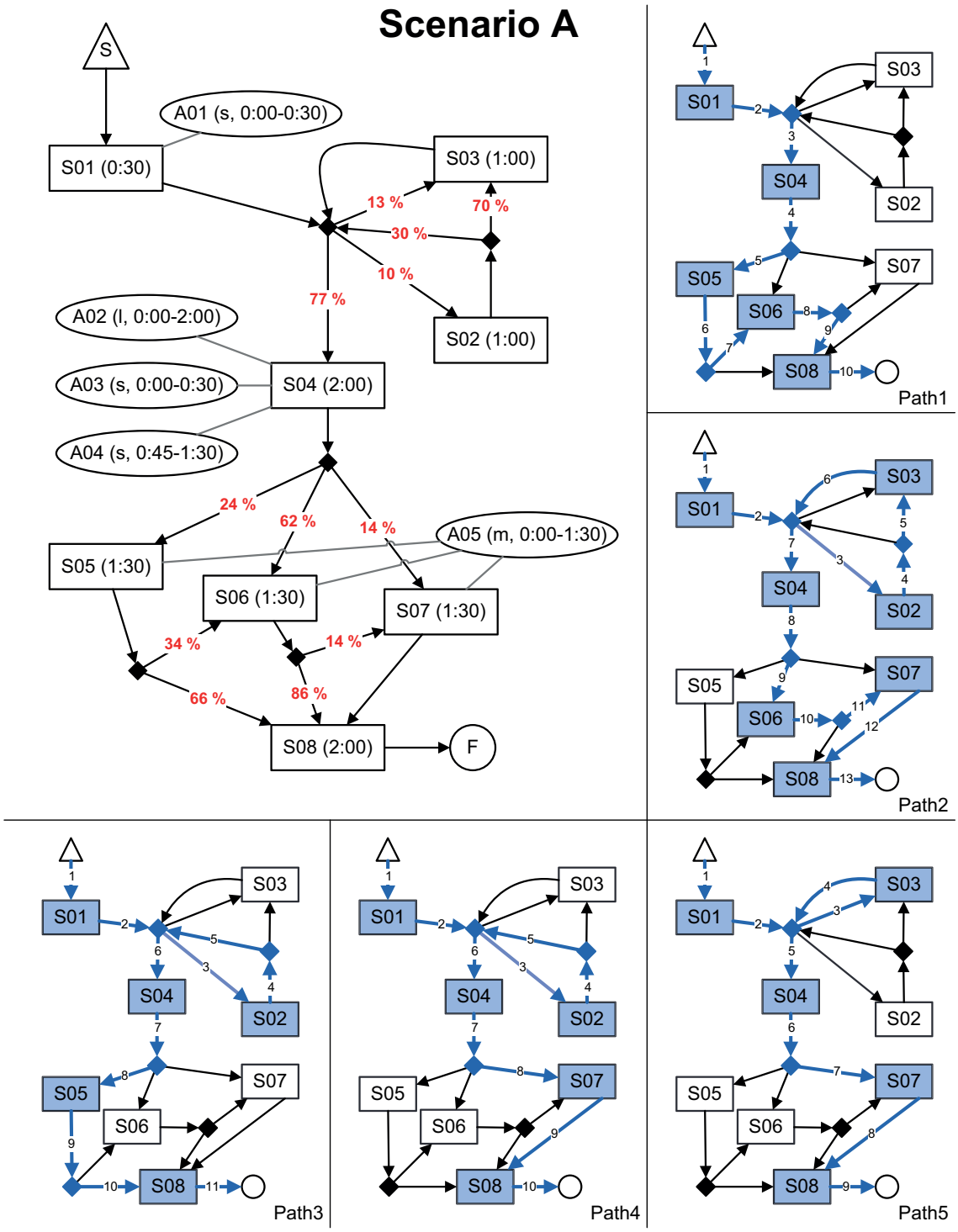


Figure F.2.: Scenario A: scene graph with probabilities and annotations (top left) and five paths through the graph.

				Settings for scenario A
Environment settings				
Environment name	Resolution	Color depth (bit)	Bandwidth (Mbit/s)	Cache (MB)
E1	1920 x 1080	24	50	1024
E2	1280 x 800	24	32	768
E3	800 x 600	8	10	256
Annotation sizes (KB)				
Annotation name	Size	E1	E2	E3
A01, A03, A04	small	12150	6000	937,5
A2	large	151875	75000	11718,75
A5	medium	60750	30000	4687,5
Scene sizes (MB)				
Scene name	Duration (sec)	E1	E2	E3
S01	30	4449,46	2197,27	343,32
S02, S03	60	8898,93	4394,53	686,65
S04, S08	120	17797,85	8789,06	1373,29
S05, S06, S07	90	13348,39	6591,80	1029,97
Path sizes and durations				
Path name	Duration (min:sec)	E1	E2	E3
Path1	7:30	66741,94	32958,98	5149,84
Path2	9:30	84539,79	41748,05	6523,13
Path3	7:00	62292,48	30761,72	4806,52
Path4	7:00	62292,48	30761,72	4806,52
Path5	7:00	62292,48	30761,72	4806,52

Table F.1.: Scenario A: settings for the environment and calculations for the annotation sizes, the scene sizes, and the path sizes and durations.

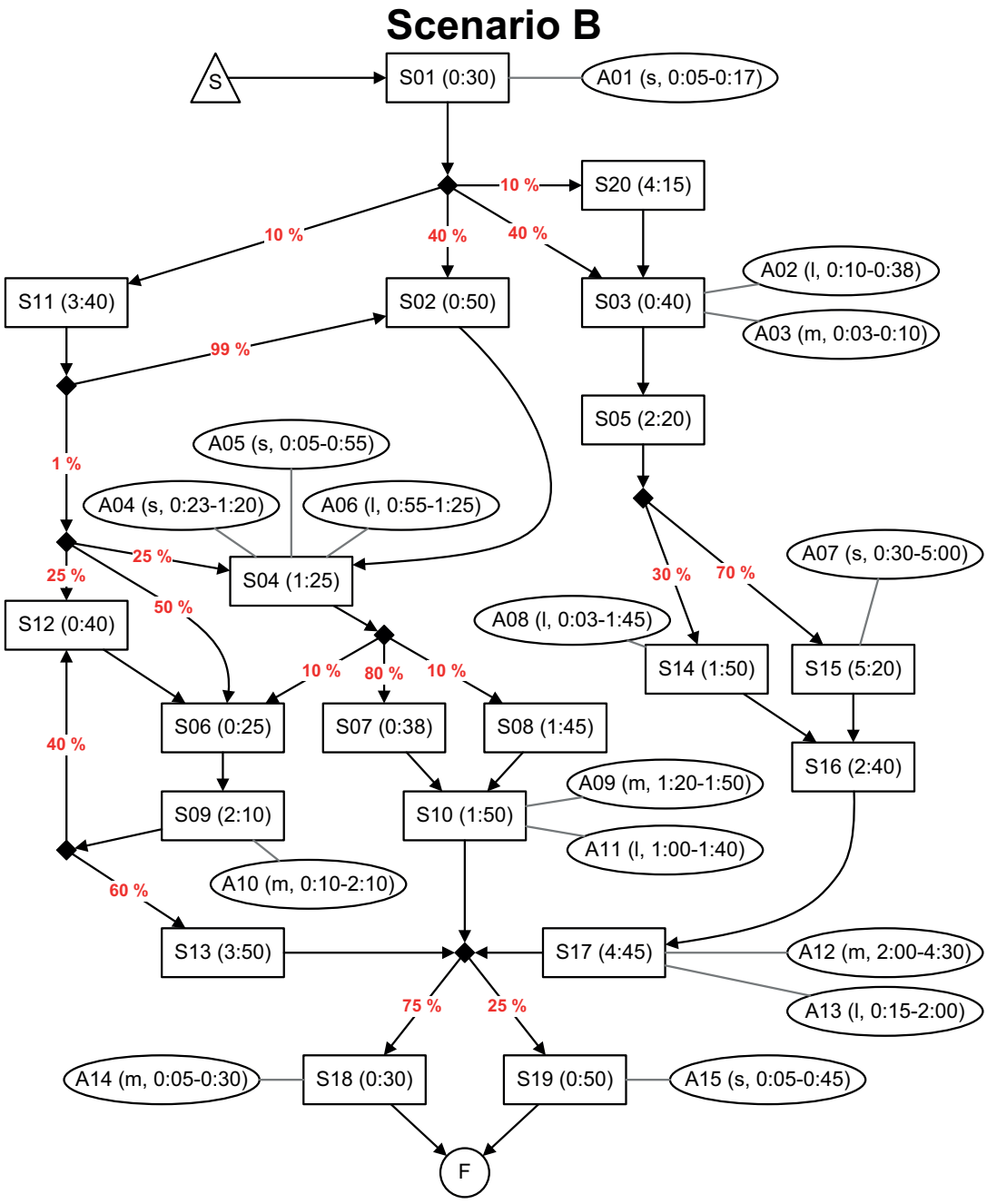


Figure F.3.: Scenario B: scene graph with probabilities and annotations.

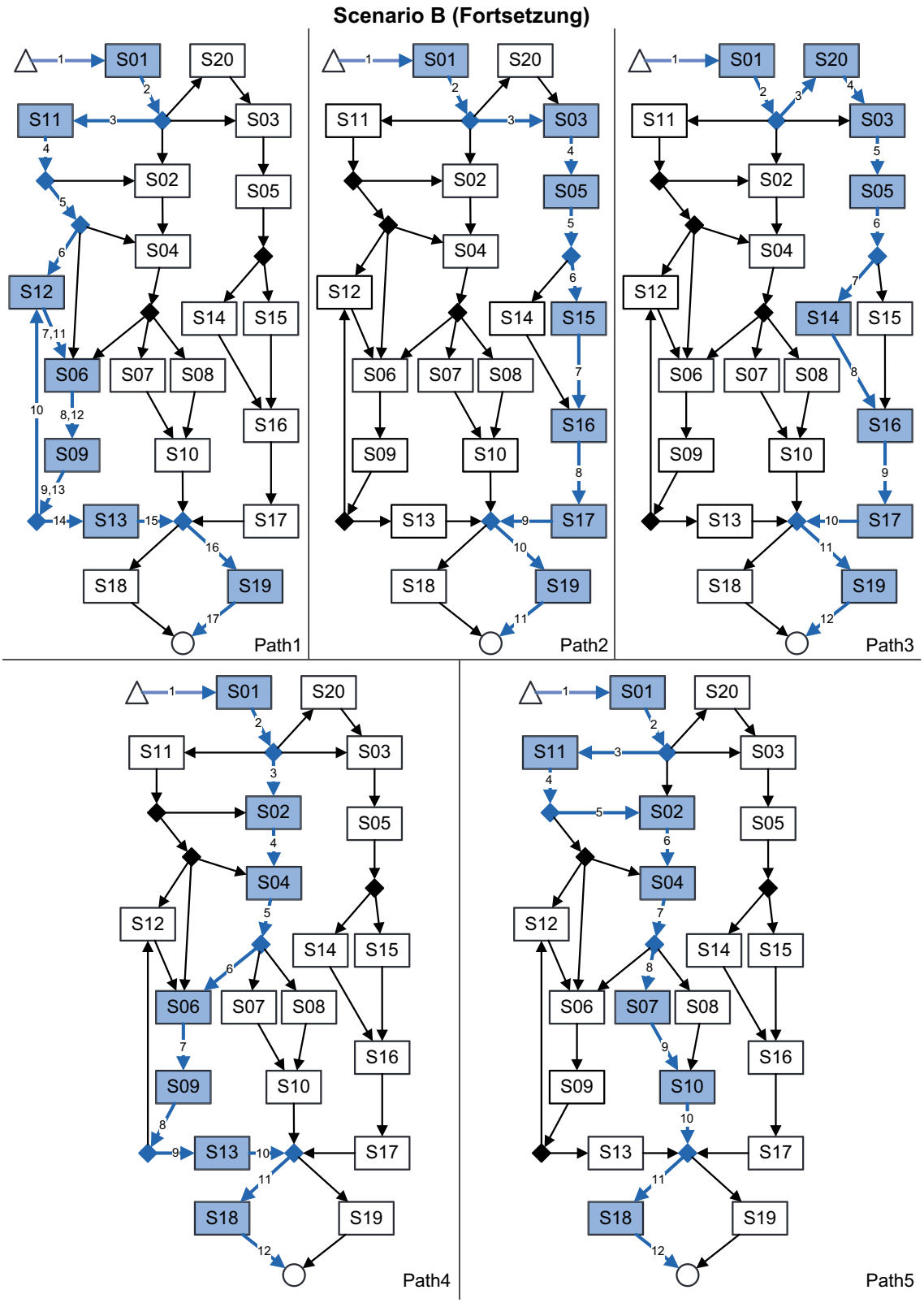


Figure F.4.: Scenario B: five paths through the graph.

Settings for scenario B				
Environment settings				
Environment name	Resolution	Color depth (bit)	Bandwidth (Mbit/s)	Cache (MB)
E1	1920 x 1080	16	50	768
E2	1680 x 1050	32	100	128
E3	1024 x 768	24	25	256
Annotation sizes (KB)				
Annotation name	Size	E1	E2	E3
A01, A04, A05, A07, A15	small	8100	13781,25	4608
A03, A09, A10, A12, A14	medium	40500	68906,25	23040
A02, A06, A08, A11, A13	large	101250	172265,625	57600
Scene sizes (MB)				
Scene name	Duration (sec)	E1	E2	E3
S01, S18	30	2966,31	5046,84	1687,50
S02, S19	50	4943,85	8411,41	2812,50
S03, S12	40	3955,08	6729,13	2250,00
S04	85	8404,54	14299,39	4781,25
S05	140	13842,77	23551,94	7875,00
S06	25	2471,92	4205,70	1406,25
S07	38	3757,32	6392,67	2137,50
S08	105	10382,08	17663,96	5906,25
S09	130	12854,00	21869,66	7312,50
S10, S14	110	10876,46	18505,10	6187,50
S11	220	21752,93	37010,19	12375,00
S13	230	22741,70	38692,47	12937,50
S15	320	31640,63	53833,01	18000,00
S16	160	15820,31	26916,50	9000,00
S17	285	28179,93	47945,02	16031,25
S20	255	25213,62	42898,18	14343,75
Path sizes and durations				
Path name	Duration (min:sec)	E1	E2	E3
Path1	15:20	90966,80	154769,90	51750,00
Path2	17:05	101348,88	172433,85	57656,25
Path3	17:50	105798,34	180004,12	60187,50
Path4	9:40	57348,63	97572,33	32625,00
Path5	9:23	55667,72	94712,45	31668,75

Table F.2.: Scenario B: settings for the environment and calculations for the annotation sizes, the scene sizes, and the path sizes and durations.

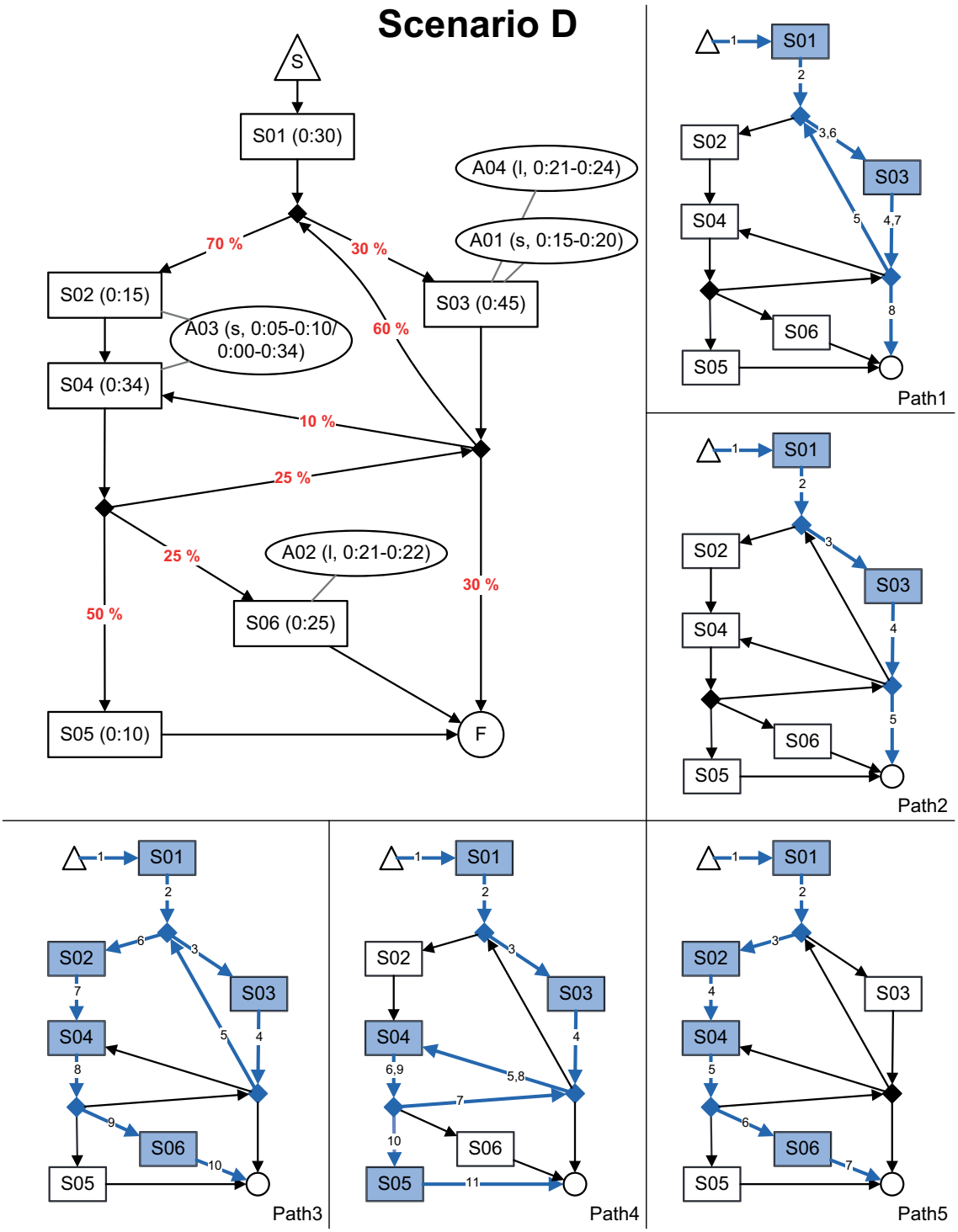


Figure F.5.: Scenario D: scene graph with probabilities and annotations (top left) and five paths through the graph.

Settings for scenario D				
Environment settings				
Environment name	Resolution	Color depth (bit)	Bandwidth (Mbit/s)	Cache (MB)
E1	1920 x 1080	32	32	1024
E2	1600 x 900	16	32	512
E3	1024 x 768	32	100	256
Annotation sizes (KB)				
Annotation name	Size	E1	E2	E3
A01	small	16200	6679,69	6144
A03	medium	81000	28125	30720
A02, A04	large	202500	70312,5	76800
Scene sizes (MB)				
Scene name	Duration (sec)	E1	E2	E3
S01	30	5932,62	2059,94	2250,00
S02	15	2966,31	1029,97	1125,00
S03	45	8898,93	3089,90	3375,00
S04	34	6723,63	2334,59	2550,00
S05	10	1977,54	686,65	750,00
S06	25	4943,85	1716,61	1875,00
Path sizes and durations				
Path name	Duration (min:sec)	E1	E2	E3
Path1	2:00	14831,54	5149,84	5625,00
Path2	1:15	14831,54	5149,84	5625,00
Path3	2:29	29465,33	10231,02	11175,00
Path4	2:33	23532,71	8171,08	8925,00
Path5	1:44	20566,41	7141,11	7800,00

Table F.3.: Scenario D: settings for the environment and calculations for the annotation sizes, the scene sizes, and the path sizes and durations.

G. Statistics for 1000 Test Runs on a Subset of Patterns and Settings

Statistics for WF_{start}

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(WF_{start})$	$max(WF_{start})$	$mean(WF_{start})$	$median(WF_{start})$	$sd(WF_{start})$
PML	$Cycle_3$	512	6	6460	7222	7175.24	7175.00	23.54
			25	4186	4943	4442.09	4442.00	22.39
			50	1472	2726	1861.25	1823.00	105.51
			100	335	690	666.35	670.00	34.97
		4096	6	3338	3352	3339.43	3339.00	1.27
			25	2103	2127	2104.58	2104.00	2.34
			50	1236	1329	1239.44	1238.00	5.17
			100	335	335	335.00	335.00	0.00
		16384	6	3339	3354	3339.42	3339.00	1.19
			25	2103	2130	2104.76	2104.00	2.31
			50	1236	1318	1239.42	1238.00	5.24
			100	335	335	335.00	335.00	0.00
PML	Sequence	512	6	6970	6993	6973.85	6974.00	1.29
			25	4968	5039	4972.99	4970.00	5.26
			50	2480	2600	2494.11	2491.00	12.27
			100	335	335	335.00	335.00	0.00
		4096	6	6969	6991	6970.43	6970.00	1.43
			25	4955	5074	4961.26	4959.00	8.46
			50	2441	2655	2454.36	2455.00	13.77
			100	335	335	335.00	335.00	0.00
		16384	6	6967	6985	6970.54	6970.00	1.32
			25	4954	5013	4961.24	4960.00	5.32
			50	2440	2689	2451.46	2445.00	19.26
			100	335	335	335.00	335.00	0.00
PML	$Sieve_3$	512	6	7360	7377	7364.22	7364.00	1.95
			25	5642	5708	5657.45	5654.00	12.72
			50	3879	3957	3899.02	3891.00	15.12
			100	1346	1604	1387.75	1384.00	31.83
		4096	6	7358	7370	7362.57	7363.00	1.74
			25	5637	5717	5644.64	5638.00	12.45
			50	3880	3962	3898.41	3900.00	17.52
			100	1134	1376	1163.72	1142.00	34.00
		16384	6	7357	7372	7362.83	7363.00	1.81
			25	5638	5685	5645.27	5638.00	9.29
			50	3880	4017	3889.42	3880.00	19.99
			100	1134	1279	1186.17	1192.50	34.57
PS	$Cycle_3$	512	6	6416	7993	7174.23	7175.00	49.17
			25	5867	6683	6562.49	6556.00	78.10
			50	3415	4480	4018.76	4034.00	79.10
			100	3600	4294	4087.28	4100.00	98.87
		4096	6	3338	3353	3339.47	3339.00	1.17
			25	2322	2341	2324.14	2324.00	2.15
			50	1639	1664	1641.59	1640.00	3.34
			100	918	933	919.23	919.00	1.33

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(WF_{start})$	$max(WF_{start})$	$mean(WF_{start})$	$median(WF_{start})$	$sd(WF_{start})$
		16384	6	3338	3352	3339.39	3339.00	0.82
			25	2322	2365	2324.29	2324.00	2.21
			50	1635	1672	1641.34	1640.00	3.67
			100	918	1000	919.64	919.00	3.41
PS	Sequence	512	6	6971	6979	6974.09	6974.00	1.23
			25	5106	5173	5116.87	5114.00	5.26
			50	2749	2907	2762.45	2759.00	11.68
			100	750	750	750.00	750.00	0.00
		4096	6	6969	6983	6970.93	6971.00	1.44
			25	5094	5180	5105.01	5103.00	6.99
			50	2707	2851	2721.97	2723.00	10.36
			100	750	750	750.00	750.00	0.00
		16384	6	6968	6976	6970.73	6971.00	1.29
			25	5094	5157	5104.05	5103.00	5.36
			50	2706	2889	2721.97	2714.00	14.23
			100	750	750	750.00	750.00	0.00
PS	$Sieve_3$	512	6	7360	7378	7364.61	7364.00	2.06
			25	6620	6693	6634.54	6631.00	13.15
			50	5780	5911	5793.32	5785.00	15.93
			100	4234	4577	4280.21	4266.00	32.75
		4096	6	7358	7371	7363.22	7363.00	1.66
			25	6608	6701	6615.23	6609.00	12.32
			50	5724	5831	5748.34	5745.00	22.47
			100	4022	4383	4048.74	4028.00	36.71
		16384	6	7358	7375	7362.96	7363.00	1.61
			25	6609	6656	6614.33	6609.00	8.16
			50	5724	5859	5742.09	5727.00	26.08
			100	4011	4165	4074.91	4081.00	43.92
PSU	$Cycle_3$	512	6	2067	2736	2443.67	2414.00	93.96
			25	967	2099	1356.97	1469.00	197.61
			50	410	883	433.84	431.00	25.42
			100	250	500	496.00	500.00	31.38
		4096	6	1083	1098	1085.13	1085.00	1.30
			25	740	786	757.15	757.00	2.58
			50	424	528	479.59	481.00	7.10
			100	250	250	250.00	250.00	0.00
		16384	6	1084	1098	1085.08	1085.00	1.01
			25	745	788	757.46	757.00	3.50
			50	421	499	476.44	481.00	10.03
			100	250	250	250.00	250.00	0.00
PSU	Sequence	512	6	2500	2500	2500.00	2500.00	0.00
			25	1656	1764	1683.64	1683.00	7.87
			50	485	725	546.60	543.00	20.25
			100	250	435	250.18	250.00	5.84
		4096	6	2500	2500	2500.00	2500.00	0.00
			25	1643	1742	1664.05	1662.00	8.50
			50	624	813	702.86	704.00	16.15
			100	250	252	250.00	250.00	0.09
		16384	6	2500	2500	2500.00	2500.00	0.00
			25	1646	1855	1665.13	1662.00	14.19
			50	589	806	694.90	701.00	21.00
			100	250	250	250.00	250.00	0.00
PSU	$Sieve_3$	512	6	2500	2500	2500.00	2500.00	0.00
			25	2184	2269	2212.13	2211.00	14.51
			50	1669	1830	1714.94	1709.00	20.39
			100	725	1209	777.30	761.00	60.38
		4096	6	2500	2500	2500.00	2500.00	0.00
			25	2183	2247	2197.12	2200.00	10.21
			50	1699	1833	1773.32	1774.50	23.38

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(WF_{start})$	$max(WF_{start})$	$mean(WF_{start})$	$median(WF_{start})$	$sd(WF_{start})$
			100	565	834	590.29	571.00	31.81
		16384	6	2500	2500	2500.00	2500.00	0.00
			25	2183	2320	2201.68	2199.00	19.51
			50	1676	1819	1751.02	1750.00	20.39
			100	565	771	606.29	617.00	29.89

Table G.1.: Statistics for 1000 test runs for the numbers of frames to wait at the beginning of a scene.

Statistics for WT_{start}

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(WT_{start})$	$max(WT_{start})$	$mean(WT_{start})$	$median(WT_{start})$	$sd(WT_{start})$
PML	$Cycle_3$	512	6	1900	2340	2238.44	2240.00	15.45
			25	180	280	199.46	200.00	4.91
			50	20	40	20.22	20.00	2.09
			100	0	0	0.00	0.00	0.00
		4096	6	1560	1580	1560.02	1560.00	0.63
			25	180	200	199.94	200.00	1.09
			50	20	40	20.02	20.00	0.63
			100	0	0	0.00	0.00	0.00
		16384	6	1560	1560	1560.00	1560.00	0.00
			25	180	200	199.52	200.00	3.06
			50	20	40	20.04	20.00	0.89
			100	0	0	0.00	0.00	0.00
PML	Sequence	512	6	3240	3260	3240.06	3240.00	1.09
			25	420	480	420.08	420.00	2.00
			50	20	20	20.00	20.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	3240	3240	3240.00	3240.00	0.00
			25	420	480	420.26	420.00	3.02
			50	20	20	20.00	20.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	3240	3240	3240.00	3240.00	0.00
			25	420	420	420.00	420.00	0.00
			50	20	20	20.00	20.00	0.00
			100	0	0	0.00	0.00	0.00
PML	$Sieve_3$	512	6	3420	3580	3464.62	3460.00	33.66
			25	600	620	600.10	600.00	1.41
			50	200	240	200.28	200.00	2.82
			100	0	20	0.02	0.00	0.63
		4096	6	3420	3460	3420.50	3420.00	4.02
			25	600	640	600.10	600.00	1.90
			50	200	220	200.06	200.00	1.09
			100	0	40	0.06	0.00	1.41
		16384	6	3420	3460	3420.54	3420.00	3.70
			25	600	640	600.18	600.00	2.44
			50	200	200	200.00	200.00	0.00

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(WT_{start})$	$max(WT_{start})$	$mean(WT_{start})$	$median(WT_{start})$	$sd(WT_{start})$
			100	0	0	0.00	0.00	0.00
PS	$Cycle_3$	512	6	1520	2600	2237.74	2240.00	42.34
			25	260	340	283.06	280.00	8.80
			50	60	120	94.88	100.00	13.52
			100	20	100	59.88	60.00	8.00
		4096	6	1560	1560	1560.00	1560.00	0.00
			25	220	240	220.04	220.00	0.89
			50	60	60	60.00	60.00	0.00
			100	20	20	20.00	20.00	0.00
		16384	6	1560	1560	1560.00	1560.00	0.00
			25	220	220	220.00	220.00	0.00
			50	60	60	60.00	60.00	0.00
			100	20	20	20.00	20.00	0.00
PS	Sequence	512	6	3240	3240	3240.00	3240.00	0.00
			25	440	460	440.04	440.00	0.89
			50	40	60	40.02	40.00	0.63
			100	20	20	20.00	20.00	0.00
		4096	6	3240	3240	3240.00	3240.00	0.00
			25	440	440	440.00	440.00	0.00
			50	40	40	40.00	40.00	0.00
			100	20	20	20.00	20.00	0.00
		16384	6	3240	3260	3240.06	3240.00	1.09
			25	440	460	440.02	440.00	0.63
			50	40	40	40.00	40.00	0.00
			100	20	20	20.00	20.00	0.00
PS	$Sieve_3$	512	6	3420	3580	3463.02	3460.00	32.04
			25	620	680	643.16	640.00	16.07
			50	220	260	220.12	220.00	1.79
			100	20	60	20.10	20.00	1.90
		4096	6	3420	3460	3420.32	3420.00	3.08
			25	620	720	640.02	640.00	15.77
			50	220	260	220.28	220.00	2.95
			100	20	60	20.08	20.00	1.55
		16384	6	3420	3460	3420.16	3420.00	2.36
			25	620	680	620.30	620.00	3.73
			50	220	280	228.32	220.00	12.02
			100	20	60	20.16	20.00	2.36
PSU	$Cycle_3$	512	6	660	1000	852.24	840.00	35.80
			25	40	100	53.40	60.00	9.58
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	480	520	517.38	520.00	6.81
			25	40	60	40.04	40.00	0.89
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	480	520	518.92	520.00	4.70
			25	40	60	40.02	40.00	0.63
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	Sequence	512	6	1160	1200	1197.90	1200.00	6.51
			25	20	80	20.32	20.00	3.33
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	1160	1200	1197.50	1200.00	7.19
			25	20	60	20.36	20.00	2.95
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	1160	1200	1198.58	1200.00	5.51
			25	20	80	20.98	20.00	6.15

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(WT_{start})$	$max(WT_{start})$	$mean(WT_{start})$	$median(WT_{start})$	$sd(WT_{start})$
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	$Sieve_3$	512	6	1080	1180	1141.58	1140.00	20.99
			25	200	220	200.04	200.00	0.89
			50	0	20	0.02	0.00	0.63
			100	0	20	0.02	0.00	0.63
		4096	6	1120	1220	1192.02	1200.00	14.00
			25	200	220	200.02	200.00	0.63
			50	0	40	0.08	0.00	1.55
			100	0	40	0.06	0.00	1.41
		16384	6	1120	1240	1193.00	1200.00	12.42
			25	200	240	200.22	200.00	2.75
			50	0	20	0.02	0.00	0.63
			100	0	0	0.00	0.00	0.00

Table G.2.: Statistics for 1000 test runs for the waiting time at the beginning of a scene.

Statistics for P_{sum}

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(P_{sum})$	$max(P_{sum})$	$mean(P_{sum})$	$median(P_{sum})$	$sd(P_{sum})$
PML	$Cycle_3$	512	6	544	848	752.79	752.00	12.11
			25	95	157	116.07	117.00	4.48
			50	7	38	20.33	22.00	4.79
			100	0	3	0.02	0.00	0.22
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PML	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PML	$Sieve_3$	512	6	0	0	0.00	0.00	0.00
			25	0	3	1.08	1.00	0.36
			50	1	5	2.83	3.00	0.74
			100	0	0	0.00	0.00	0.00

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(P_{sum})$	$max(P_{sum})$	$mean(P_{sum})$	$median(P_{sum})$	$sd(P_{sum})$
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	$Cycle_3$	512	6	542	1064	752.95	752.00	17.70
			25	106	158	128.07	121.00	9.67
			50	12	68	25.43	25.00	3.51
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	$Sieve_3$	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	$Cycle_3$	512	6	1663	2598	2064.65	2010.00	167.68
			25	277	480	340.19	367.00	42.58
			50	88	107	103.54	105.00	3.28
			100	3	8	5.00	5.00	0.19
		4096	6	1114	1117	1115.03	1115.00	0.23
			25	198	203	200.07	200.00	0.32
			50	59	64	60.31	60.00	0.57
			100	0	0	0.00	0.00	0.00
		16384	6	1112	1116	1115.01	1115.00	0.21
			25	199	204	200.09	200.00	0.37
			50	59	64	60.42	60.00	0.69
			100	0	0	0.00	0.00	0.00
PSU	Sequence	512	6	2227	2233	2230.34	2230.00	0.63
			25	397	408	400.46	400.00	0.79
			50	126	130	128.83	129.00	0.48
			100	5	6	5.01	5.00	0.08

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(P_{sum})$	$max(P_{sum})$	$mean(P_{sum})$	$median(P_{sum})$	$sd(P_{sum})$
		4096	6	2227	2233	2230.09	2230.00	0.40
			25	398	407	400.26	400.00	0.78
			50	118	127	120.29	120.00	0.74
			100	0	0	0.00	0.00	0.00
		16384	6	2227	2232	2230.03	2230.00	0.37
			25	399	405	400.22	400.00	0.62
			50	119	128	120.33	120.00	0.84
			100	0	0	0.00	0.00	0.00
PSU	$Sieve_3$	512	6	2229	2240	2236.82	2237.00	1.07
			25	402	414	407.18	407.00	1.02
			50	129	136	131.50	131.00	0.77
			100	0	9	1.48	1.00	0.66
		4096	6	2223	2233	2230.09	2230.00	0.95
			25	399	406	400.37	400.00	0.70
			50	118	128	120.56	120.00	1.11
			100	0	0	0.00	0.00	0.00
		16384	6	2225	2233	2230.05	2230.00	0.80
			25	397	406	400.37	400.00	0.75
			50	119	128	120.38	120.00	0.87
			100	0	0	0.00	0.00	0.00

Table G.3.: Statistics for 1000 test runs for the number of pauses during playback.

Statistics for $DL_{not\ watched}$

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(DL_{not\ watched})$	$max(DL_{not\ watched})$	$mean(DL_{not\ watched})$	$median(DL_{not\ watched})$	$sd(DL_{not\ watched})$
PML	$Cycle_3$	512	6	118	1395	127.21	117.50	70.33
			25	332	1128	761.23	785.00	93.31
			50	850	3395	1849.49	1980.00	292.63
			100	800	2700	2622.44	2637.50	168.18
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PML	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(DL_{not\ watched})$	$max(DL_{not\ watched})$	$mean(DL_{not\ watched})$	$median(DL_{not\ watched})$	$sd(DL_{not\ watched})$
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PML	$Sieve_3$	512	6	1150	1235	1191.78	1192.50	7.69
			25	2402	2700	2645.00	2657.50	63.40
			50	4665	5140	4986.66	5025.00	71.10
			100	10022	11550	11315.17	11335.00	193.25
		4096	6	1192	1235	1192.80	1192.50	3.55
			25	2325	2700	2670.66	2700.00	57.38
			50	4725	5300	5089.97	5090.00	100.03
			100	11330	12600	12445.90	12557.50	189.93
		16384	6	1192	1235	1192.67	1192.50	2.68
			25	2488	2700	2666.81	2700.00	46.35
			50	4410	5250	5136.89	5175.00	107.15
			100	11840	12600	12312.33	12257.50	204.06
PS	$Cycle_3$	512	6	58	1692	123.51	117.50	80.53
			25	788	2562	1387.15	1240.00	245.33
			50	1195	5450	3284.92	3462.50	481.86
			100	10292	12962	12023.55	12070.00	348.44
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	$Sieve_3$	512	6	1150	1235	1191.73	1192.50	6.54
			25	4610	4950	4895.13	4907.50	64.04
			50	9018	9688	9639.55	9687.50	83.74
			100	16535	18292	18076.80	18165.00	178.74
		4096	6	1192	1235	1192.76	1192.50	3.28
			25	4532	4950	4924.49	4950.00	53.54
			50	9398	9900	9794.90	9815.00	109.83
			100	17488	19332	19221.84	19332.50	185.28
		16384	6	1192	1235	1192.67	1192.50	2.68
			25	4780	4950	4927.01	4950.00	37.55
			50	9230	9900	9824.91	9900.00	128.42
			100	18578	19418	19117.50	19085.00	216.56
PSU	$Cycle_3$	512	6	25	392	349.40	357.50	45.58
			25	395	1335	675.50	700.00	189.35
			50	650	2168	1176.57	1230.00	183.67
			100	0	1975	1949.07	1975.00	208.04
		4096	6	0	0	0.00	0.00	0.00

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(DL_{not\ watched})$	$max(DL_{not\ watched})$	$mean(DL_{not\ watched})$	$median(DL_{not\ watched})$	$sd(DL_{not\ watched})$
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	$Sieve_3$	512	6	75	200	142.50	150.00	24.33
			25	1820	2275	2081.30	2067.50	81.50
			50	3942	4762	4551.22	4567.50	103.64
			100	7535	10052	9787.48	9875.00	306.64
		4096	6	125	250	213.95	225.00	18.14
			25	1978	2318	2250.40	2235.00	63.25
			50	3855	4652	4148.97	4145.00	127.48
			100	9592	11092	10972.38	11092.50	180.32
		16384	6	125	225	215.97	225.00	15.83
			25	1565	2318	2225.07	2235.00	110.86
			50	3935	4780	4249.05	4230.00	113.10
			100	9932	11092	10874.38	10793.75	172.24

Table G.4.: Statistics for 1000 test runs for the data volume of downloaded but not watched elements.

Statistics for *RDLV*

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(RDLV)$	$max(RDLV)$	$mean(RDLV)$	$median(RDLV)$	$sd(RDLV)$
PML	$Cycle_3$	512	6	6622	9848	8762.73	8762.50	103.30
			25	6885	9062	7193.87	7202.50	80.63
			50	8955	12672	9726.74	9897.50	451.97
			100	8698	11332	11249.45	11270.00	221.51
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(RDLV)$	$max(RDLV)$	$mean(RDLV)$	$median(RDLV)$	$sd(RDLV)$
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PML	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PML	$Sieve_3$	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	$Cycle_3$	512	6	5380	13968	8758.34	8762.50	262.53
			25	8988	11280	9517.34	9372.50	268.83
			50	9580	14168	12502.40	12790.00	748.23
			100	21852	25412	24437.02	24520.00	503.47
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PS	$Sieve_3$	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(RDLV)$	$max(RDLV)$	$mean(RDLV)$	$median(RDLV)$	$sd(RDLV)$
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	$Cycle_3$	512	6	4912	11838	8112.00	7720.00	1175.80
			25	6275	13488	8618.32	9577.50	1475.63
			50	7438	9160	8147.86	8220.00	198.41
			100	7490	10075	10011.14	10047.50	290.66
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	Sequence	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
PSU	$Sieve_3$	512	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
		4096	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00
	3	16384	6	0	0	0.00	0.00	0.00
			25	0	0	0.00	0.00	0.00
			50	0	0	0.00	0.00	0.00
			100	0	0	0.00	0.00	0.00

Table G.5.: Statistics for 1000 test runs for the data volume of repeatedly downloaded elements.

Statistics for DLV

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(DLV)$	$max(DLV)$	$mean(DLV)$	$median(DLV)$	$sd(DLV)$
PML	$Cycle_3$	512	6	16998	20222	19137.72	19137.50	103.30
			25	17260	19438	17568.87	17577.50	80.63
			50	19330	23048	20101.74	20272.50	451.97

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(DLV)$	$max(DLV)$	$mean(DLV)$	$median(DLV)$	$sd(DLV)$
			100	19072	21708	21624.45	21645.00	221.51
		4096	6	10375	10375	10375.00	10375.00	0.00
			25	10375	10375	10375.00	10375.00	0.00
			50	10375	10375	10375.00	10375.00	0.00
			100	10375	10375	10375.00	10375.00	0.00
		16384	6	10375	10375	10375.00	10375.00	0.00
			25	10375	10375	10375.00	10375.00	0.00
			50	10375	10375	10375.00	10375.00	0.00
			100	10375	10375	10375.00	10375.00	0.00
PML	Sequence	512	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
		4096	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
		16384	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
PML	$Sieve_3$	512	6	21900	21985	21941.78	21942.50	7.69
			25	23152	23450	23395.00	23407.50	63.40
			50	25415	25890	25736.66	25775.00	71.10
			100	30772	32300	32065.17	32085.00	193.25
		4096	6	21942	21985	21942.80	21942.50	3.55
			25	23075	23450	23420.66	23450.00	57.38
			50	25475	26050	25839.97	25840.00	100.03
			100	32080	33350	33195.89	33307.50	189.93
		16384	6	21942	21985	21942.67	21942.50	2.68
			25	23238	23450	23416.81	23450.00	46.35
			50	25160	26000	25886.89	25925.00	107.15
			100	32590	33350	33062.33	33007.50	204.06
PS	$Cycle_3$	512	6	15755	24342	19133.34	19137.50	262.53
			25	19362	21655	19892.34	19747.50	268.83
			50	19955	24542	22877.40	23165.00	748.23
			100	32228	35788	34812.02	34895.00	503.47
		4096	6	10375	10375	10375.00	10375.00	0.00
			25	10375	10375	10375.00	10375.00	0.00
			50	10375	10375	10375.00	10375.00	0.00
			100	10375	10375	10375.00	10375.00	0.00
		16384	6	10375	10375	10375.00	10375.00	0.00
			25	10375	10375	10375.00	10375.00	0.00
			50	10375	10375	10375.00	10375.00	0.00
			100	10375	10375	10375.00	10375.00	0.00
PS	Sequence	512	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
		4096	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
		16384	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
PS	$Sieve_3$	512	6	21900	21985	21941.74	21942.50	6.54
			25	25360	25700	25645.13	25657.50	64.04
			50	29768	30438	30389.55	30437.50	83.74

Continued

StrategyPrefPl	Pattern	Cache	Bandwidth	$min(DLV)$	$max(DLV)$	$mean(DLV)$	$median(DLV)$	$sd(DLV)$
			100	37285	39042	38826.80	38915.00	178.74
		4096	6	21942	21985	21942.76	21942.50	3.28
			25	25282	25700	25674.49	25700.00	53.54
			50	30148	30650	30544.90	30565.00	109.83
			100	38238	40082	39971.84	40082.50	185.28
		16384	6	21942	21985	21942.67	21942.50	2.68
			25	25530	25700	25677.01	25700.00	37.55
			50	29980	30650	30574.91	30650.00	128.42
			100	39328	40168	39867.50	39835.00	216.56
PSU	$Cycle_3$	512	6	15288	22212	18487.00	18095.00	1175.80
			25	16650	23862	18993.31	19952.50	1475.63
			50	17812	19535	18522.86	18595.00	198.41
			100	17865	20450	20386.14	20422.50	290.66
		4096	6	10375	10375	10375.00	10375.00	0.00
			25	10375	10375	10375.00	10375.00	0.00
			50	10375	10375	10375.00	10375.00	0.00
			100	10375	10375	10375.00	10375.00	0.00
		16384	6	10375	10375	10375.00	10375.00	0.00
			25	10375	10375	10375.00	10375.00	0.00
			50	10375	10375	10375.00	10375.00	0.00
			100	10375	10375	10375.00	10375.00	0.00
PSU	Sequence	512	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
		4096	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
		16384	6	20750	20750	20750.00	20750.00	0.00
			25	20750	20750	20750.00	20750.00	0.00
			50	20750	20750	20750.00	20750.00	0.00
			100	20750	20750	20750.00	20750.00	0.00
PSU	$Sieve_3$	512	6	20825	20950	20892.50	20900.00	24.33
			25	22570	23025	22831.30	22817.50	81.50
			50	24692	25512	25301.22	25317.50	103.64
			100	28285	30802	30537.48	30625.00	306.64
		4096	6	20875	21000	20963.95	20975.00	18.14
			25	22728	23068	23000.40	22985.00	63.25
			50	24605	25402	24898.97	24895.00	127.48
			100	30342	31842	31722.38	31842.50	180.32
		16384	6	20875	20975	20965.97	20975.00	15.83
			25	22315	23068	22975.07	22985.00	110.86
			50	24685	25530	24999.04	24980.00	113.10
			100	30682	31842	31624.38	31543.75	172.24

Table G.6.: Statistics for 1000 test runs for the download volume.

H. Multiple Comparison Test after Kruskal-Wallis Test for the 1000 Test Runs

Multiple comparison test after Kruskal-Wallis for WF_{start}

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
Sequence - 512 - 6	2164.995	0	PML-PS	116.86	160.73	FALSE
			PML-PSU	1441.57	160.73	TRUE
			PS-PSU	1558.43	160.73	TRUE
Sequence - 512 - 25	2677.185	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 512 - 50	2668.157	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 512 - 100	2998.63	0	PML-PS	1001.00	160.89	TRUE
			PML-PSU	999.50	160.77	TRUE
			PS-PSU	2000.50	160.77	TRUE
Sequence - 4096 - 6	2154.333	0	PML-PS	226.43	160.84	TRUE
			PML-PSU	1387.79	160.76	TRUE
			PS-PSU	1614.21	160.76	TRUE
Sequence - 4096 - 25	2669.937	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 4096 - 50	2667.3	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 4096 - 100	2997.879	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 16384 - 6	2121.21	0	PML-PS	94.06	160.73	FALSE
			PML-PSU	1452.97	160.73	TRUE
			PS-PSU	1547.03	160.73	TRUE
Sequence - 16384 - 25	2672.192	0	PML-PS	1000.00	160.89	TRUE
			PML-PSU	1001.50	160.77	TRUE
			PS-PSU	2001.50	160.77	TRUE
Sequence - 16384 - 50	2669.649	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 16384 - 100	2999	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 512 - 6	2069.669	0	PML-PS	98.70	160.73	FALSE
			PML-PSU	1450.65	160.73	TRUE
			PS-PSU	1549.35	160.73	TRUE
$Cycle_3$ - 512 - 25	2667.728	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 512 - 50	2672.151	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 512 - 100	2853.322	0	PML-PS	1010.82	160.73	TRUE
			PML-PSU	978.35	160.73	TRUE
			PS-PSU	1989.18	160.73	TRUE
$Cycle_3$ - 4096 - 6	2315.547	0	PML-PS	57.82	160.73	FALSE

Continued

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
			PML-PSU	1471.09	160.73	TRUE
			PS-PSU	1528.91	160.73	TRUE
$Cycle_3$ - 4096 - 25	2699.431	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 4096 - 50	2707.019	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 4096 - 100	2965.229	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 16384 - 6	2331.844	0	PML-PS	16.45	160.73	FALSE
			PML-PSU	1491.77	160.73	TRUE
			PS-PSU	1508.23	160.73	TRUE
$Cycle_3$ - 16384 - 25	2693.95	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 16384 - 50	2701.257	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 16384 - 100	2952.095	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 512 - 6	2139.743	0	PML-PS	145.32	160.73	FALSE
			PML-PSU	1427.34	160.73	TRUE
			PS-PSU	1572.66	160.73	TRUE
$Sieve_3$ - 512 - 25	2668.06	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 512 - 50	2668.723	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 512 - 100	2668.667	0	PML-PS	1000.00	160.89	TRUE
			PML-PSU	1001.50	160.77	TRUE
			PS-PSU	2001.50	160.77	TRUE
$Sieve_3$ - 4096 - 6	2138.881	0	PML-PS	223.14	160.84	TRUE
			PML-PSU	1389.43	160.76	TRUE
			PS-PSU	1612.57	160.76	TRUE
$Sieve_3$ - 4096 - 25	2737.48	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 4096 - 50	2670.868	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 4096 - 100	2671.942	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 16384 - 6	2105.099	0	PML-PS	66.36	160.73	FALSE
			PML-PSU	1466.82	160.73	TRUE
			PS-PSU	1533.18	160.73	TRUE
$Sieve_3$ - 16384 - 25	2709.024	0	PML-PS	1000.00	160.89	TRUE
			PML-PSU	1001.50	160.77	TRUE
			PS-PSU	2001.50	160.77	TRUE
$Sieve_3$ - 16384 - 50	2720.126	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 16384 - 100	2666.588	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE

Table H.1.: Multiple comparison test after Kruskal-Wallis for 1000 test runs for the numbers of frames to wait at the beginning of a scene.

Multiple comparison test after Kruskal-Wallis for WT_{start}

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
Sequence - 512 - 6	2949.247	0	PML-PS	3.00	160.73	FALSE
			PML-PSU	1501.50	160.73	TRUE
			PS-PSU	1498.50	160.73	TRUE
Sequence - 512 - 25	2989.327	0	PML-PS	997.00	160.73	TRUE
			PML-PSU	1001.50	160.73	TRUE
			PS-PSU	1998.50	160.73	TRUE
Sequence - 512 - 50	2998.626	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 512 - 100	3002	0	PML-PS	1501.50	160.89	TRUE
			PML-PSU	0.00	160.77	FALSE
			PS-PSU	1501.50	160.77	TRUE
Sequence - 4096 - 6	2950.808	0	PML-PS	0.00	160.84	FALSE
			PML-PSU	1501.00	160.76	TRUE
			PS-PSU	1501.00	160.76	TRUE
Sequence - 4096 - 25	2974.189	0	PML-PS	988.00	160.73	TRUE
			PML-PSU	1006.00	160.73	TRUE
			PS-PSU	1994.00	160.73	TRUE
Sequence - 4096 - 50	2999	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 4096 - 100	2999	0	PML-PS	1500.00	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1500.00	160.73	TRUE
Sequence - 16384 - 6	2962.644	0	PML-PS	3.00	160.73	FALSE
			PML-PSU	1498.50	160.73	TRUE
			PS-PSU	1501.50	160.73	TRUE
Sequence - 16384 - 25	2990.362	0	PML-PS	1000.00	160.89	TRUE
			PML-PSU	1001.50	160.77	TRUE
			PS-PSU	2001.50	160.77	TRUE
Sequence - 16384 - 50	2999	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
Sequence - 16384 - 100	2999	0	PML-PS	1500.00	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1500.00	160.73	TRUE
$Cycle_3$ - 512 - 6	2842.722	0	PML-PS	22.73	160.73	FALSE
			PML-PSU	1488.64	160.73	TRUE
			PS-PSU	1511.36	160.73	TRUE
$Cycle_3$ - 512 - 25	2857.191	0	PML-PS	999.13	160.73	TRUE
			PML-PSU	1000.43	160.73	TRUE
			PS-PSU	1999.57	160.73	TRUE
$Cycle_3$ - 512 - 50	2951.859	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 512 - 100	2937.91	0	PML-PS	1500.00	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1500.00	160.73	TRUE
$Cycle_3$ - 4096 - 6	2941.584	0	PML-PS	1.00	160.73	FALSE
			PML-PSU	1500.50	160.73	TRUE
			PS-PSU	1499.50	160.73	TRUE
$Cycle_3$ - 4096 - 25	2996.385	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 4096 - 50	2998.626	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 4096 - 100	2999	0	PML-PS	1500.00	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1500.00	160.73	TRUE
$Cycle_3$ - 16384 - 6	2974.558	0	PML-PS	0.00	160.73	FALSE
			PML-PSU	1500.00	160.73	TRUE

Continued

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
			PS-PSU	1500.00	160.73	TRUE
$Cycle_3$ - 16384 - 25	2989.872	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 16384 - 50	2998.252	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Cycle_3$ - 16384 - 100	2999	0	PML-PS	1500.00	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1500.00	160.73	TRUE
$Sieve_3$ - 512 - 6	2069.855	0	PML-PS	31.16	160.73	FALSE
			PML-PSU	1515.58	160.73	TRUE
			PS-PSU	1484.42	160.73	TRUE
$Sieve_3$ - 512 - 25	2888.842	0	PML-PS	998.74	160.73	TRUE
			PML-PSU	1000.63	160.73	TRUE
			PS-PSU	1999.37	160.73	TRUE
$Sieve_3$ - 512 - 50	2974.941	0	PML-PS	986.06	160.73	TRUE
			PML-PSU	1006.97	160.73	TRUE
			PS-PSU	1993.03	160.73	TRUE
$Sieve_3$ - 512 - 100	2991.542	0	PML-PS	1500.00	160.89	TRUE
			PML-PSU	0.00	160.77	FALSE
			PS-PSU	1500.01	160.77	TRUE
$Sieve_3$ - 4096 - 6	2841.633	0	PML-PS	5.03	160.84	FALSE
			PML-PSU	1503.51	160.76	TRUE
			PS-PSU	1498.49	160.76	TRUE
$Sieve_3$ - 4096 - 25	2905.483	0	PML-PS	997.64	160.73	TRUE
			PML-PSU	1001.18	160.73	TRUE
			PS-PSU	1998.82	160.73	TRUE
$Sieve_3$ - 4096 - 50	2989.72	0	PML-PS	997.03	160.73	TRUE
			PML-PSU	1001.48	160.73	TRUE
			PS-PSU	1998.51	160.73	TRUE
$Sieve_3$ - 4096 - 100	2972.669	0	PML-PS	1495.51	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1495.51	160.73	TRUE
$Sieve_3$ - 16384 - 6	2847.434	0	PML-PS	17.95	160.73	FALSE
			PML-PSU	1508.98	160.73	TRUE
			PS-PSU	1491.02	160.73	TRUE
$Sieve_3$ - 16384 - 25	2982.384	0	PML-PS	991.06	160.89	TRUE
			PML-PSU	1005.97	160.77	TRUE
			PS-PSU	1997.03	160.77	TRUE
$Sieve_3$ - 16384 - 50	2912.528	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 16384 - 100	2996.515	0	PML-PS	1500.00	160.73	TRUE
			PML-PSU	0.00	160.73	FALSE
			PS-PSU	1500.00	160.73	TRUE

Table H.2.: Multiple comparison test after Kruskal-Wallis for 1000 test runs for the waiting times at the beginning of a scene.

Multiple comparison test after Kruskal-Wallis for P_{sum}

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
$Cycle_3$ - 512 - 6	2191.005	0	PML-PS	177.84	160.73	TRUE
			PML-PSU	1411.08	160.73	TRUE
			PS-PSU	1588.92	160.73	TRUE

Continued

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
$Cycle_3$ - 512 - 25	2652.18	0	PML-PS	924.91	160.73	TRUE
			PML-PSU	1962.46	160.73	TRUE
			PS-PSU	1037.54	160.73	TRUE
$Cycle_3$ - 512 - 50	2466.531	0	PML-PS	751.45	160.73	TRUE
			PML-PSU	1875.73	160.73	TRUE
			PS-PSU	1124.27	160.73	TRUE
$Cycle_3$ - 512 - 100	2983.636	0	PML-PS	6.01	160.73	FALSE
			PML-PSU	1496.98	160.73	TRUE
			PS-PSU	1502.99	160.73	TRUE

Table H.3.: Multiple comparison test after Kruskal-Wallis for 1000 test runs for the number of pauses during playback.

Multiple comparison test after Kruskal-Wallis for $DL_{not\ watched}$

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
$Cycle_3$ - 512 - 6	2608.105	0	PML-PS	34.60	160.73	FALSE
			PML-PSU	1417.78	160.73	TRUE
			PS-PSU	1452.38	160.73	TRUE
$Cycle_3$ - 512 - 25	2215.333	0	PML-PS	1102.44	160.73	TRUE
			PML-PSU	687.09	160.73	TRUE
			PS-PSU	1789.53	160.73	TRUE
$Cycle_3$ - 512 - 50	2462.881	0	PML-PS	959.44	160.73	TRUE
			PML-PSU	953.44	160.73	TRUE
			PS-PSU	1912.88	160.73	TRUE
$Cycle_3$ - 512 - 100	2827.429	0	PML-PS	1010.84	160.73	TRUE
			PML-PSU	978.32	160.73	TRUE
			PS-PSU	1989.16	160.73	TRUE
$Sieve_3$ - 512 - 6	2758.992	0	PML-PS	0.91	160.73	FALSE
			PML-PSU	1500.45	160.73	TRUE
			PS-PSU	1499.55	160.73	TRUE
$Sieve_3$ - 512 - 25	2691.092	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 512 - 50	2705.484	0	PML-PS	1000.09	160.73	TRUE
			PML-PSU	999.81	160.73	TRUE
			PS-PSU	1999.91	160.73	TRUE
$Sieve_3$ - 512 - 100	2672.491	0	PML-PS	1000.13	160.89	TRUE
			PML-PSU	1001.24	160.77	TRUE
			PS-PSU	2001.37	160.77	TRUE
$Sieve_3$ - 4096 - 6	2866.119	0	PML-PS	1.00	160.84	FALSE
			PML-PSU	1501.50	160.76	TRUE
			PS-PSU	1500.50	160.76	TRUE
$Sieve_3$ - 4096 - 25	2764.833	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 4096 - 50	2674.984	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 4096 - 100	2730.385	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 16384 - 6	2882.661	0	PML-PS	0.00	160.73	FALSE
			PML-PSU	1500.00	160.73	TRUE
			PS-PSU	1500.00	160.73	TRUE
$Sieve_3$ - 16384 - 25	2735.654	0	PML-PS	1000.00	160.89	TRUE

Continued

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
			PML-PSU	1001.50	160.77	TRUE
			PS-PSU	2001.50	160.77	TRUE
$Sieve_3$ - 16384 - 50	2705.012	0	PML-PS	1000.08	160.73	TRUE
			PML-PSU	999.83	160.73	TRUE
			PS-PSU	1999.92	160.73	TRUE
$Sieve_3$ - 16384 - 100	2674.711	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE

Table H.4.: Multiple comparison test after Kruskal-Wallis for 1000 test runs for the data volume of not watched elements.

Multiple comparison test after Kruskal-Wallis for *RDLV*

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
$Cycle_3$ - 512 - 6	1496.247	0	PML-PS	32.92	160.73	FALSE
			PML-PSU	1220.49	160.73	TRUE
			PS-PSU	1187.57	160.73	TRUE
$Cycle_3$ - 512 - 25	968.488	0	PML-PS	1159.00	160.73	TRUE
			PML-PSU	841.97	160.73	TRUE
			PS-PSU	317.03	160.73	TRUE
$Cycle_3$ - 512 - 50	2667.449	0	PML-PS	994.37	160.73	TRUE
			PML-PSU	1002.76	160.73	TRUE
			PS-PSU	1997.12	160.73	TRUE
$Cycle_3$ - 512 - 100	2737.764	0	PML-PS	1010.82	160.73	TRUE
			PML-PSU	978.35	160.73	TRUE
			PS-PSU	1989.18	160.73	TRUE

Table H.5.: Multiple comparison test after Kruskal-Wallis for 1000 test runs for the data volume of repeatedly downloaded elements.

Multiple comparison test after Kruskal-Wallis for *DLV*

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
$Cycle_3$ - 512 - 6	1496.247	0	PML-PS	32.92	160.73	FALSE
			PML-PSU	1220.49	160.73	TRUE
			PS-PSU	1187.57	160.73	TRUE
$Cycle_3$ - 512 - 25	968.488	0	PML-PS	1159.00	160.73	TRUE
			PML-PSU	841.97	160.73	TRUE
			PS-PSU	317.03	160.73	TRUE
$Cycle_3$ - 512 - 50	2667.449	0	PML-PS	994.37	160.73	TRUE
			PML-PSU	1002.76	160.73	TRUE
			PS-PSU	1997.12	160.73	TRUE
$Cycle_3$ - 512 - 100	2737.764	0	PML-PS	1010.82	160.73	TRUE
			PML-PSU	978.35	160.73	TRUE
			PS-PSU	1989.18	160.73	TRUE
$Sieve_3$ - 512 - 6	2758.992	0	PML-PS	0.91	160.73	FALSE
			PML-PSU	1500.45	160.73	TRUE

Continued

Pattern - Cache size - Bandwidth	H-Value	p-Value	Strategy A vs. Strategy B	Observed difference	Critical difference	Difference
			PS-PSU	1499.55	160.73	TRUE
$Sieve_3$ - 512 - 25	2691.092	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 512 - 50	2705.484	0	PML-PS	1000.09	160.73	TRUE
			PML-PSU	999.81	160.73	TRUE
			PS-PSU	1999.91	160.73	TRUE
$Sieve_3$ - 512 - 100	2672.491	0	PML-PS	1000.13	160.89	TRUE
			PML-PSU	1001.24	160.77	TRUE
			PS-PSU	2001.37	160.77	TRUE
$Sieve_3$ - 4096 - 6	2866.119	0	PML-PS	1.00	160.84	FALSE
			PML-PSU	1501.50	160.76	TRUE
			PS-PSU	1500.50	160.76	TRUE
$Sieve_3$ - 4096 - 25	2764.833	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 4096 - 50	2674.984	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 4096 - 100	2730.385	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE
$Sieve_3$ - 16384 - 6	2882.661	0	PML-PS	0.00	160.73	FALSE
			PML-PSU	1500.00	160.73	TRUE
			PS-PSU	1500.00	160.73	TRUE
$Sieve_3$ - 16384 - 25	2735.654	0	PML-PS	1000.00	160.89	TRUE
			PML-PSU	1001.50	160.77	TRUE
			PS-PSU	2001.50	160.77	TRUE
$Sieve_3$ - 16384 - 50	2705.012	0	PML-PS	1000.08	160.73	TRUE
			PML-PSU	999.83	160.73	TRUE
			PS-PSU	1999.92	160.73	TRUE
$Sieve_3$ - 16384 - 100	2674.711	0	PML-PS	1000.00	160.73	TRUE
			PML-PSU	1000.00	160.73	TRUE
			PS-PSU	2000.00	160.73	TRUE

Table H.6.: Multiple comparison test after Kruskal-Wallis for 1000 test runs for the download volume.

I. Detailed/Further Statistics and Evaluation Results

I.1. Further Statistics

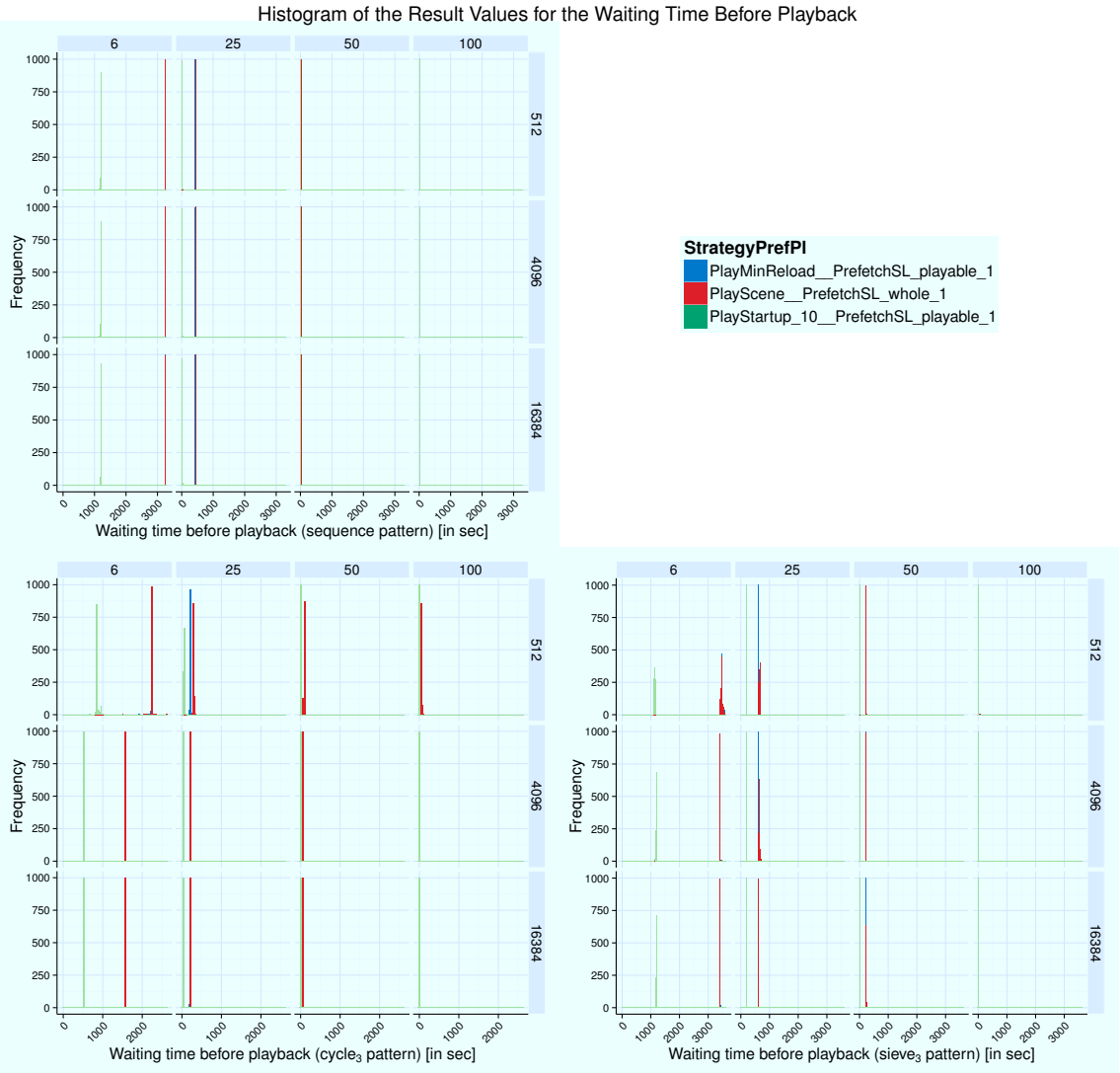


Figure I.1.: Statistics - distribution of the waiting times before playback: sequence pattern (top left), cycle pattern (bottom left), and sieve pattern (bottom right); $binwidth = 25$.

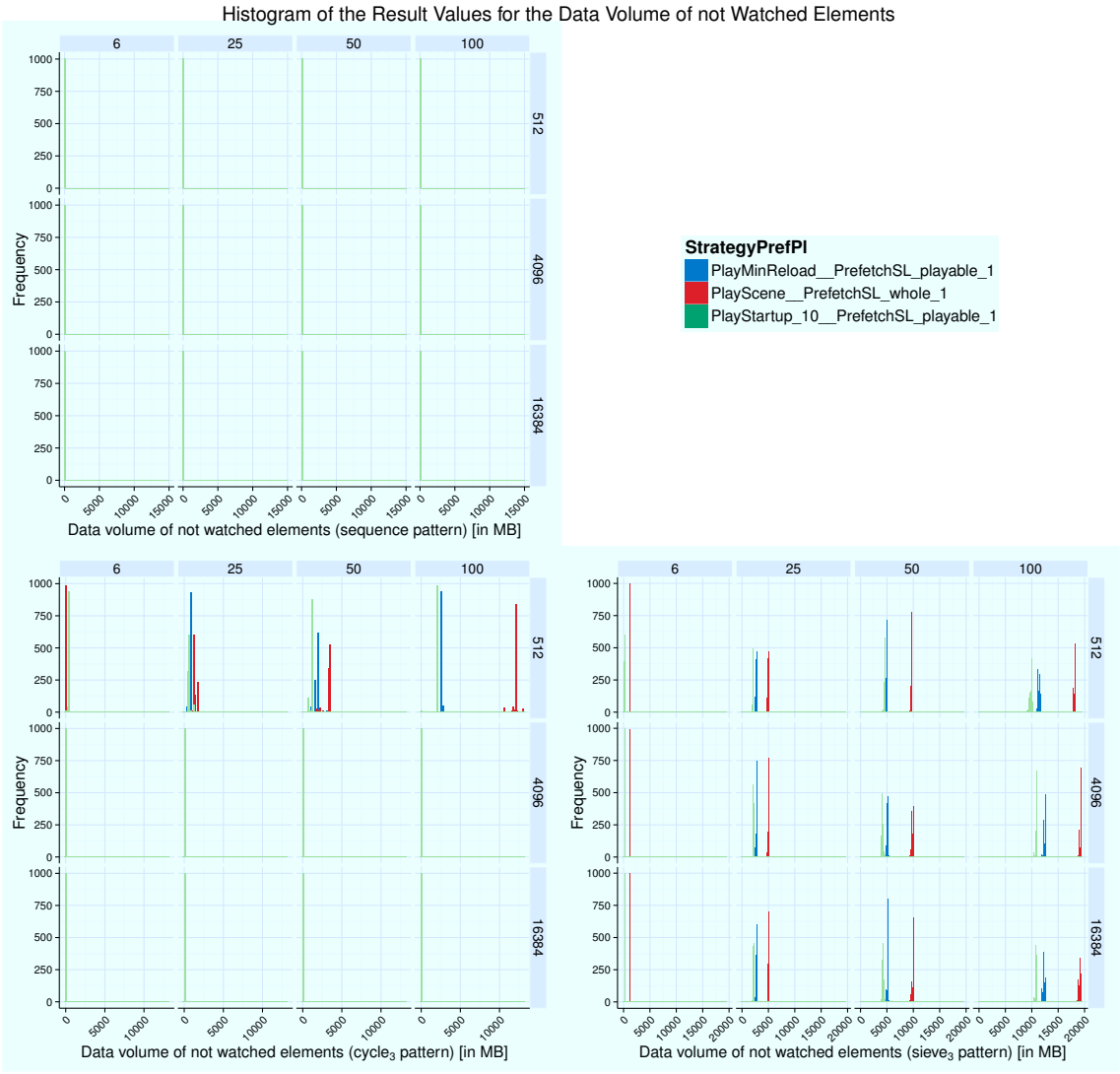


Figure I.2.: Statistics - distribution of the data volume of elements not watched: sequence pattern (top left), cycle pattern (bottom left), and sieve pattern (bottom right); $binwidth = 150$.

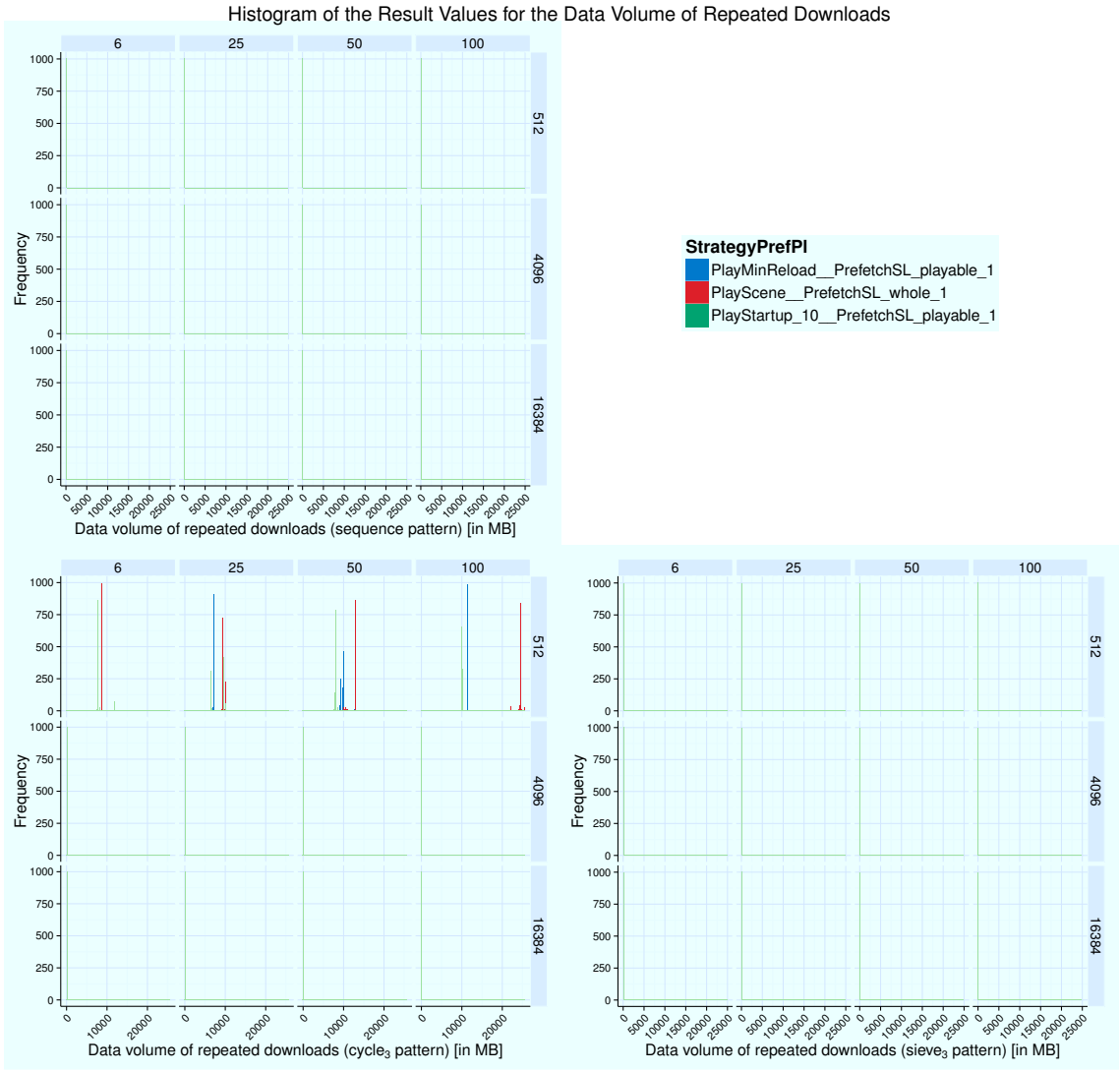


Figure I.3.: Statistics - distribution of the data volume of repeatedly downloaded elements: sequence pattern (top left), cycle pattern (bottom left), and sieve pattern (bottom right); $binwidth = 150$.

I.2. Evaluation of the Pre-fetch Strategies and Start Times

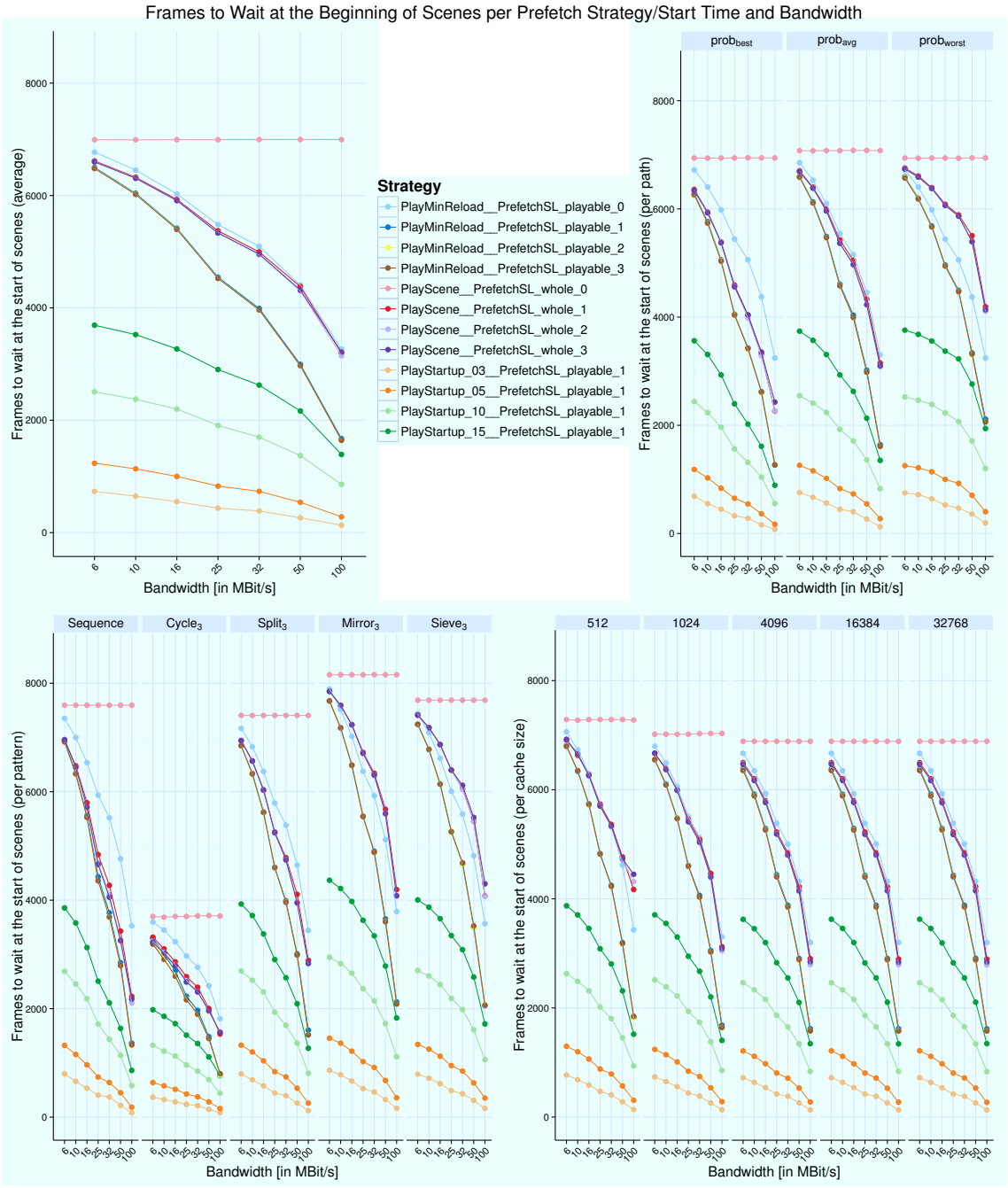


Figure I.4.: Evaluation of the pre-fetch strategies (detailed results) - frames to wait before playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

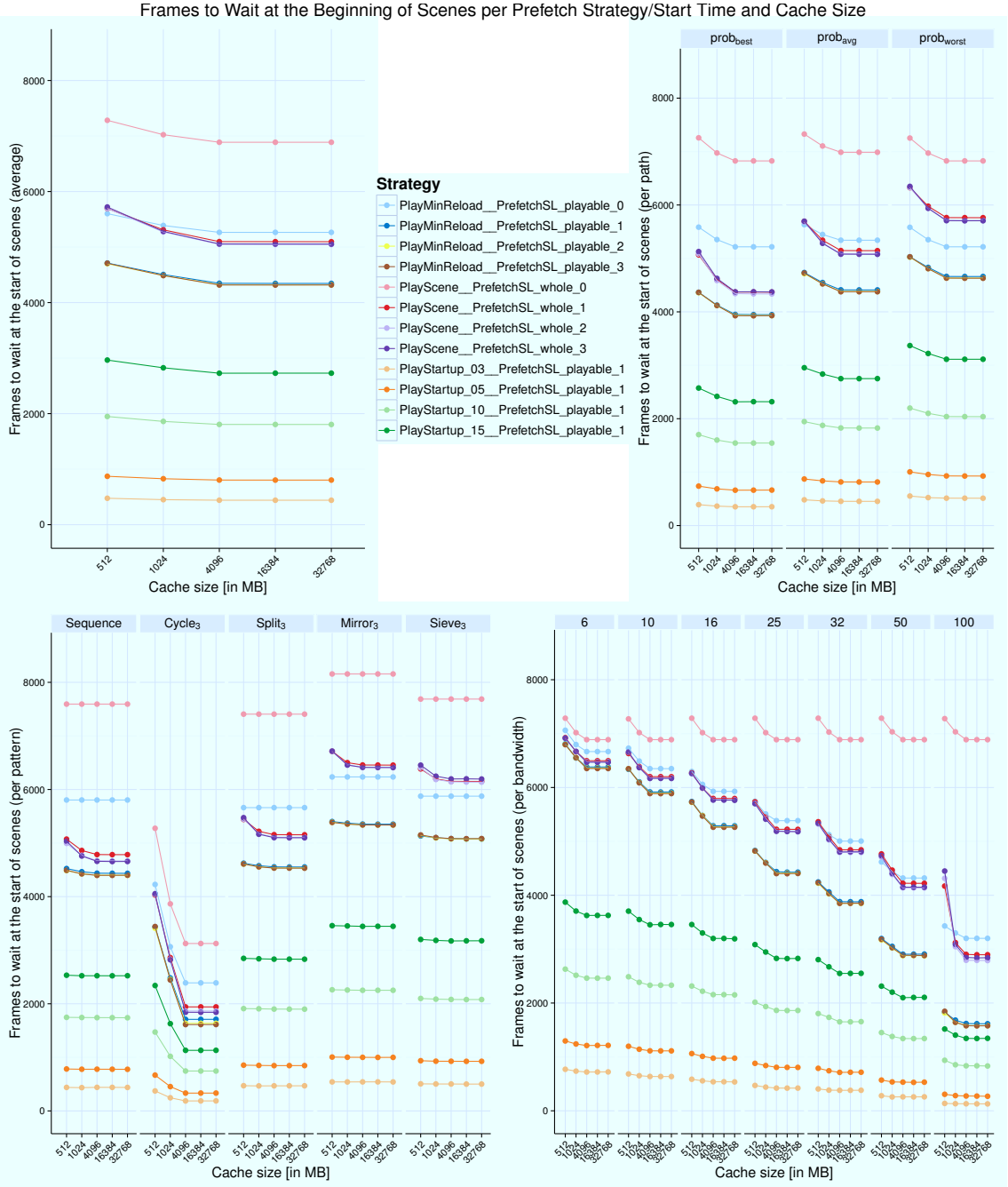


Figure I.5.: Evaluation of the pre-fetch strategies (detailed results) - frames to wait before playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

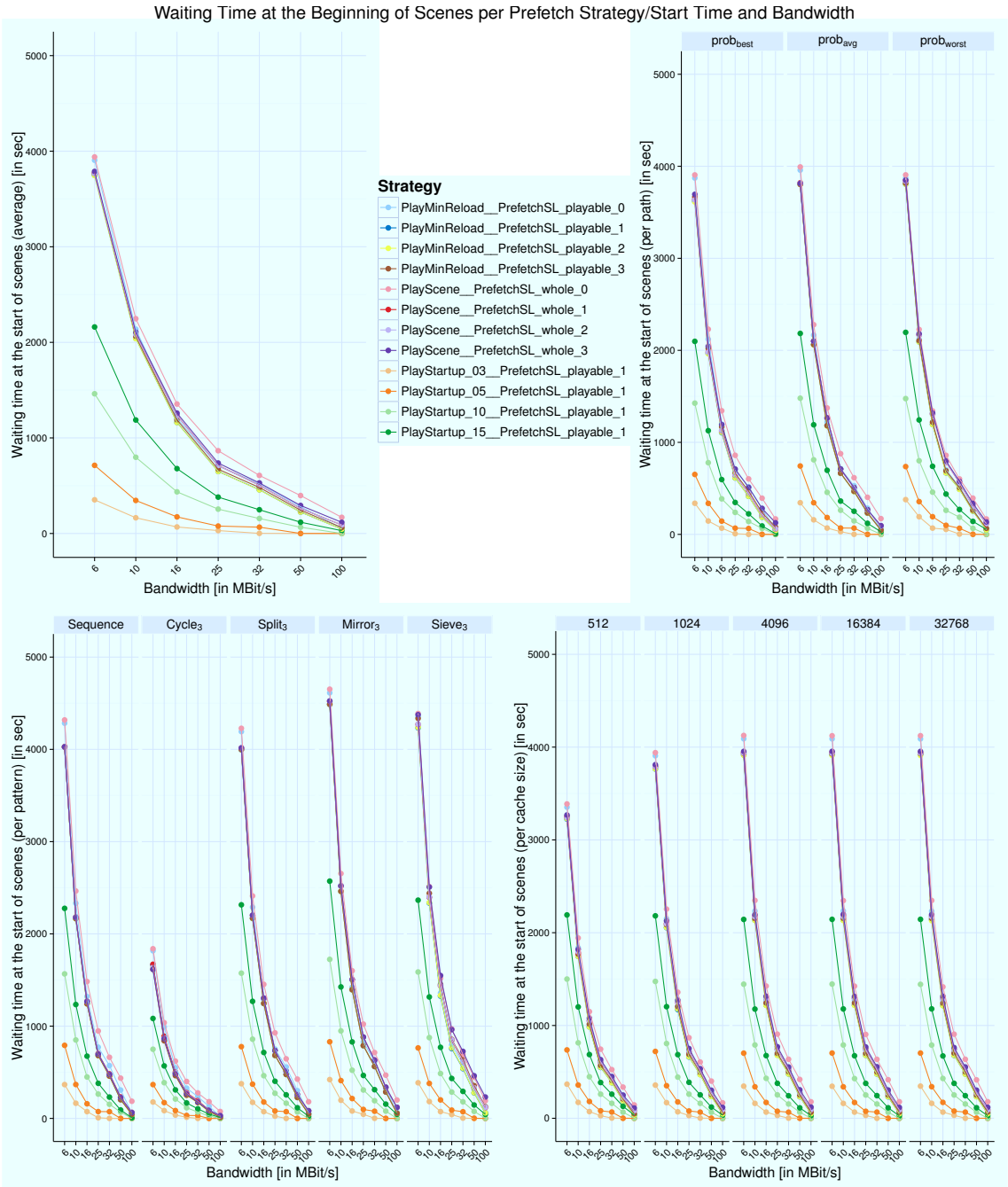


Figure I.6.: Evaluation of the pre-fetch strategies (detailed results) - waiting time before playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

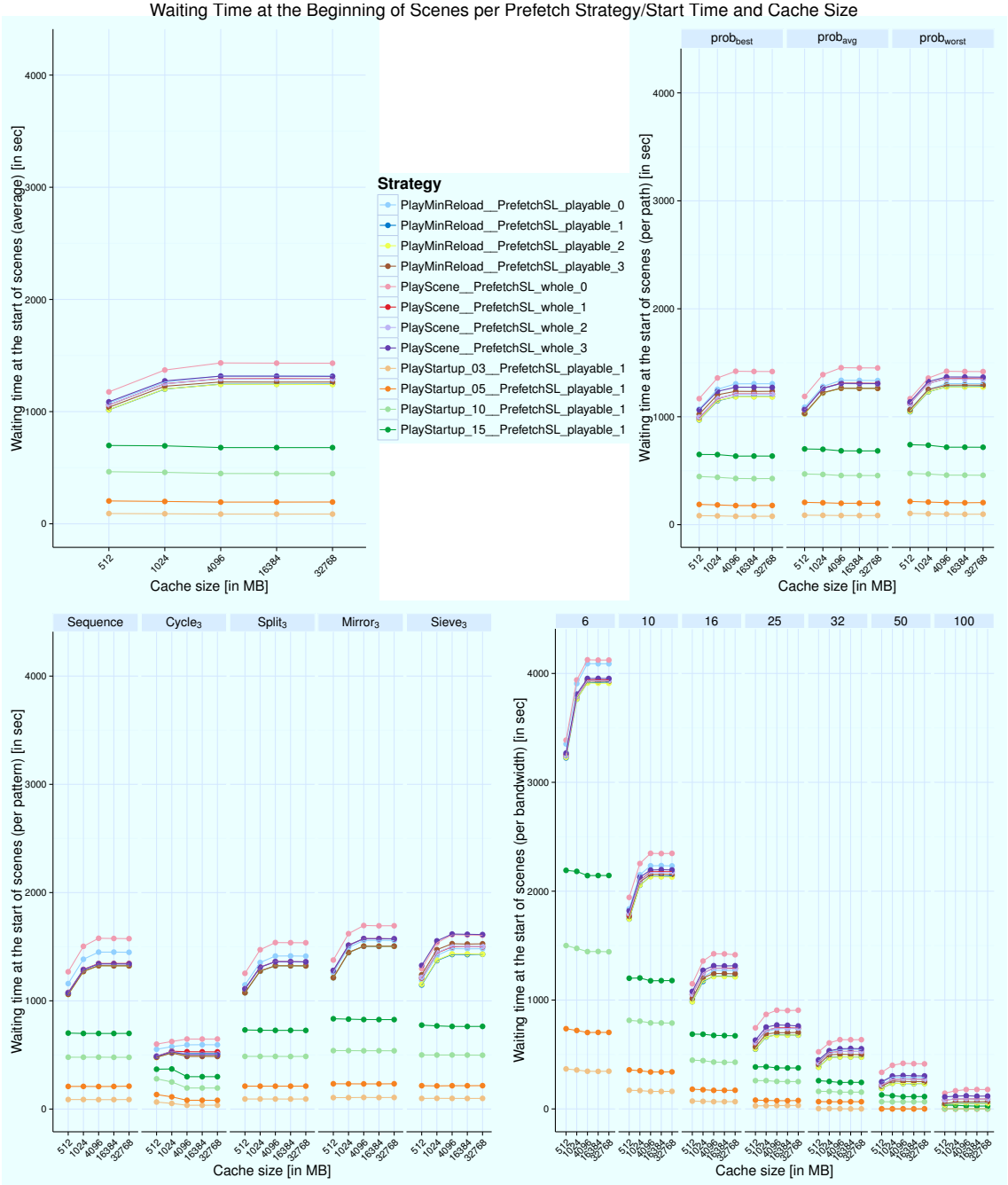


Figure I.7.: Evaluation of the pre-fetch strategies (detailed results) - waiting time before playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

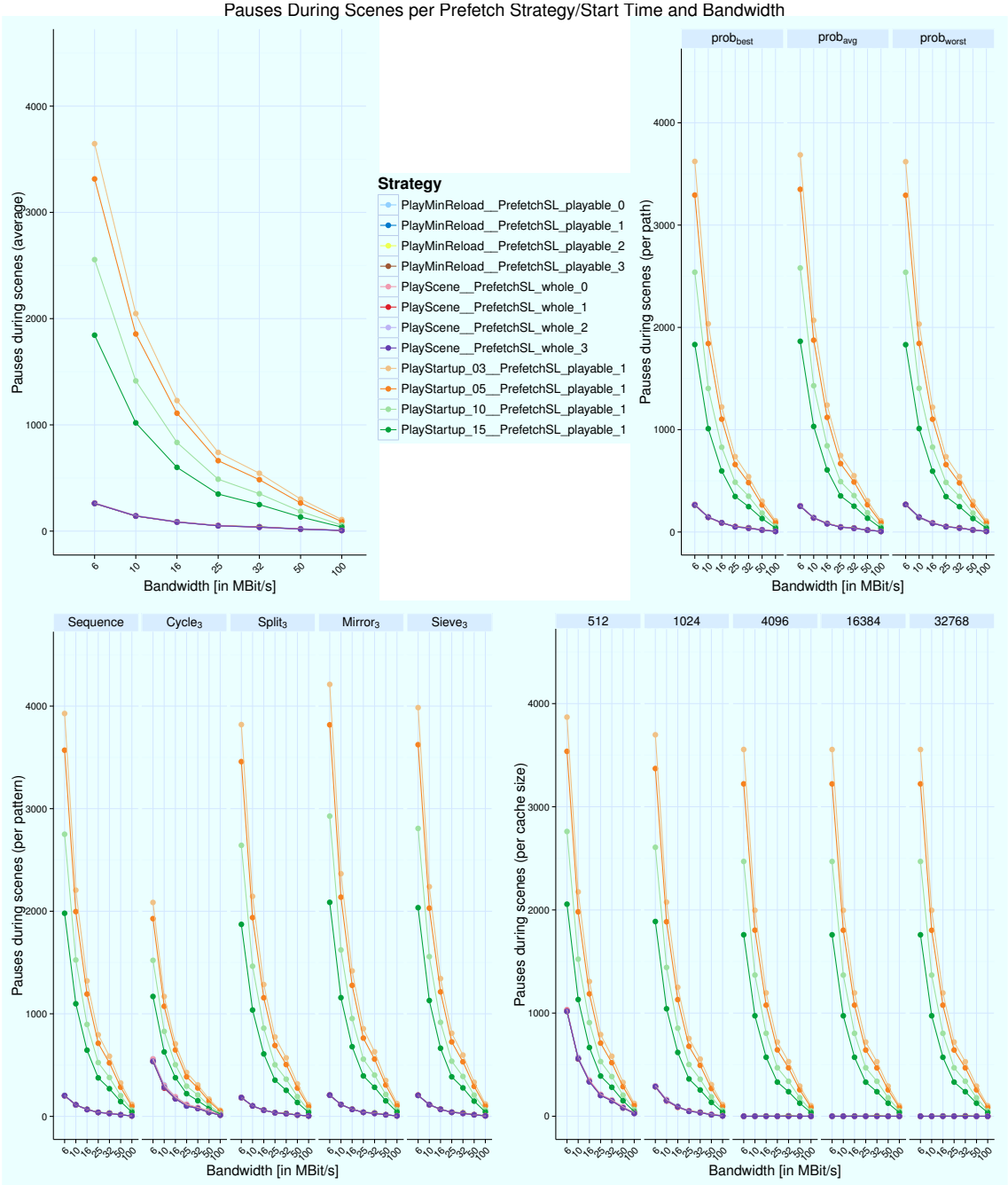


Figure I.8.: Evaluation of the pre-fetch strategies (detailed results) - pauses during playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

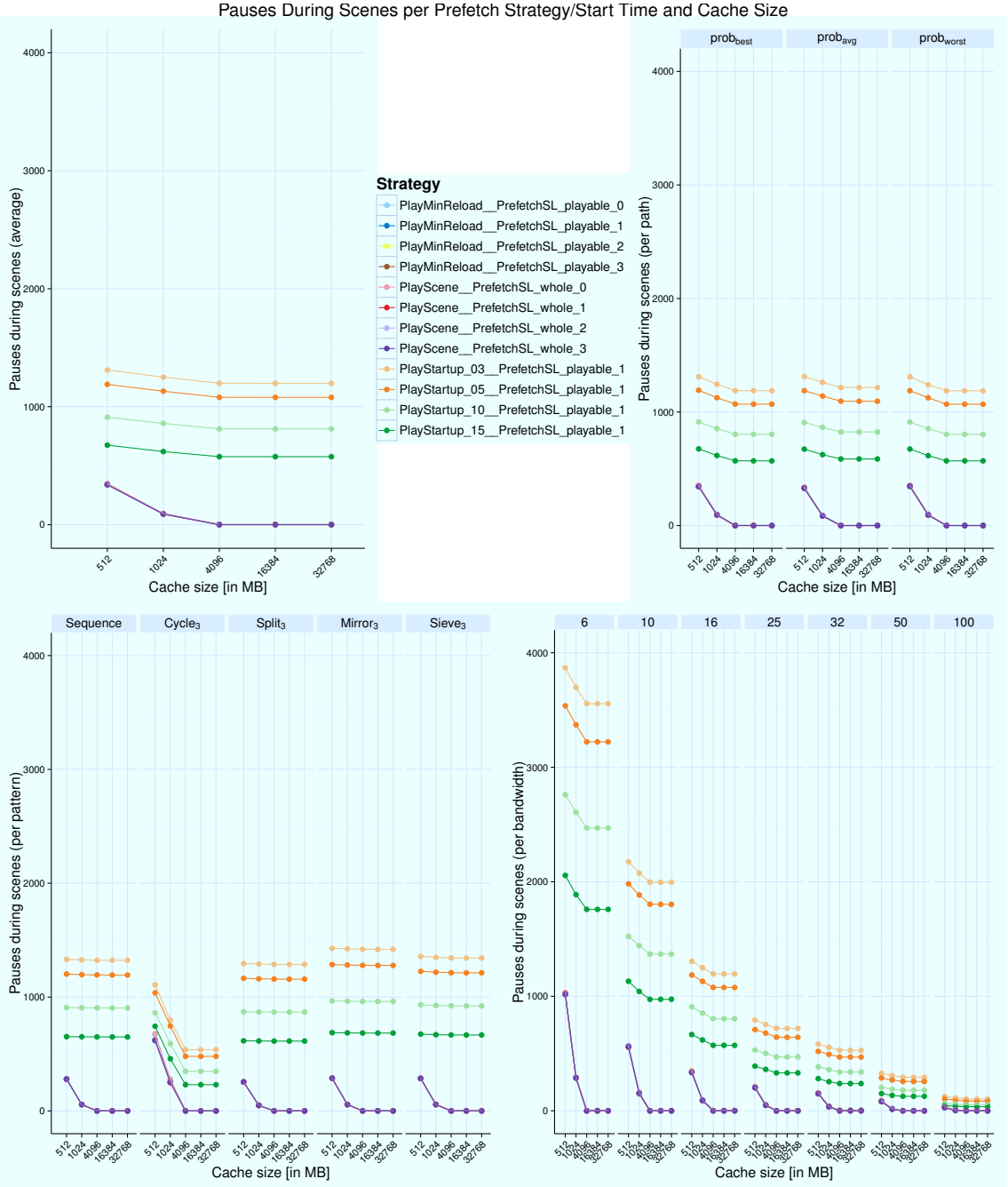


Figure I.9.: Evaluation of the pre-fetch strategies (detailed results) - pauses during playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

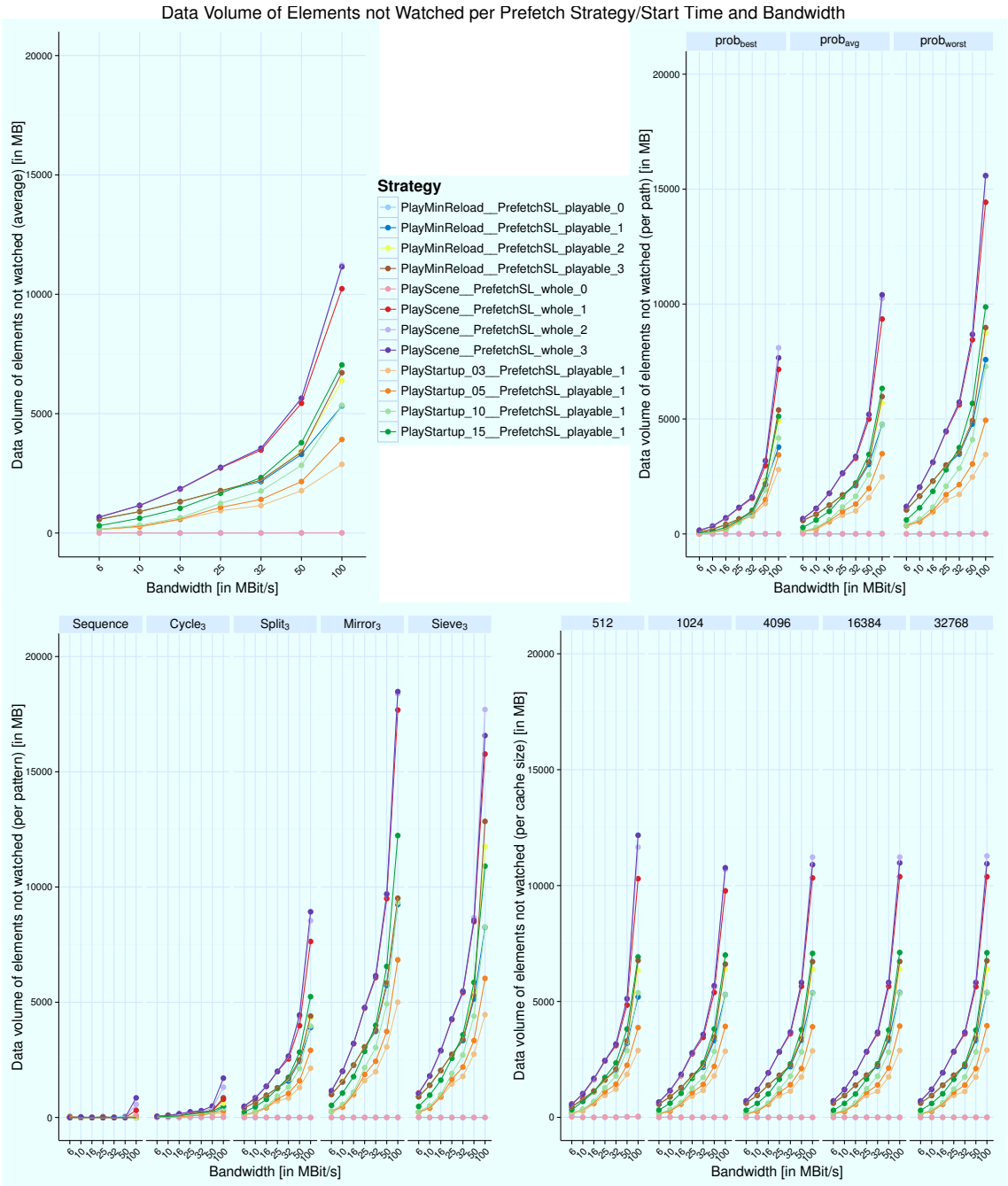


Figure I.10.: Evaluation of the pre-fetch strategies (detailed results) - data volume of elements not watched for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

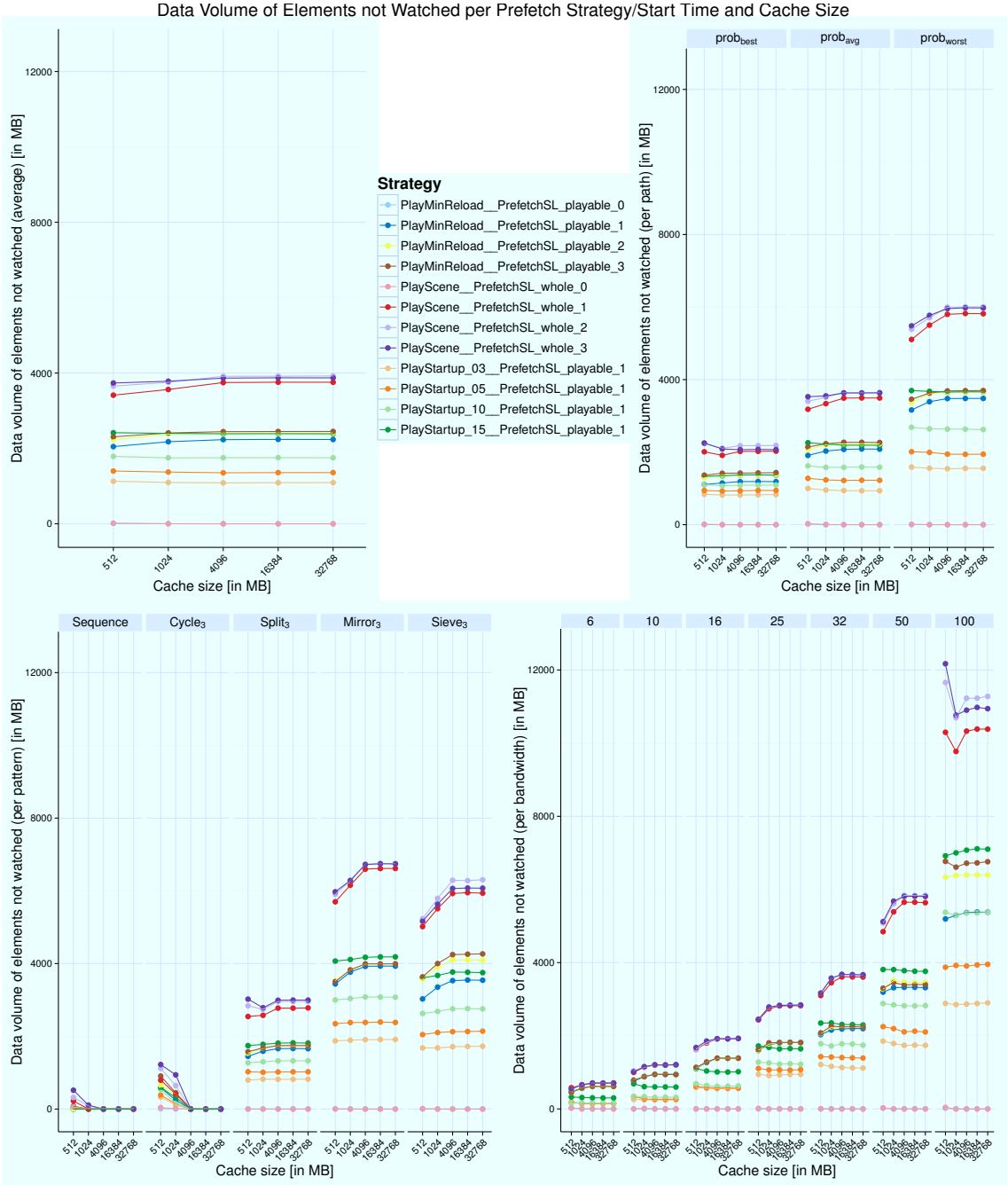


Figure I.11.: Evaluation of the pre-fetch strategies (detailed results) - data volume of elements not watched for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

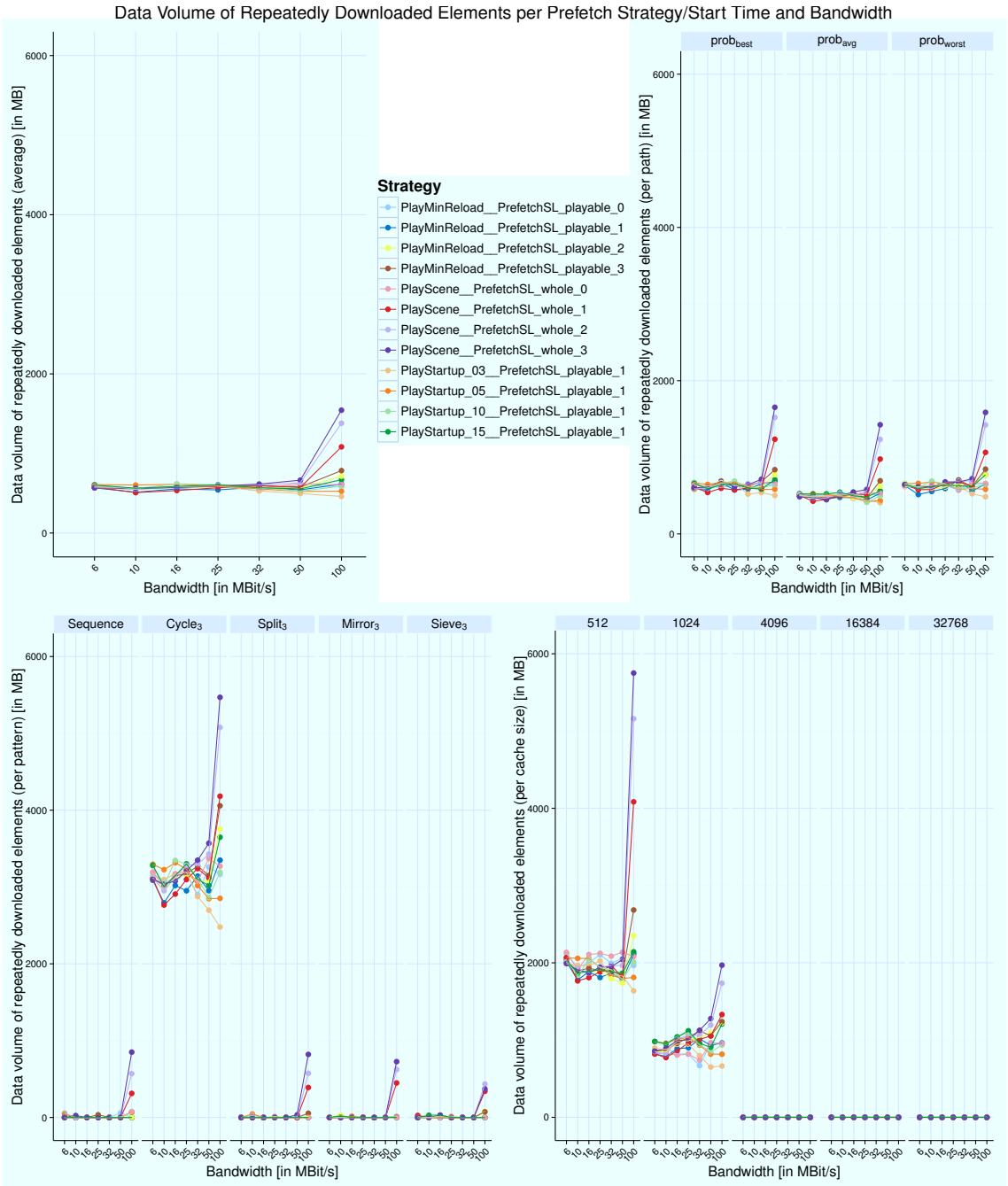


Figure I.12.: Evaluation of the pre-fetch strategies (detailed results) - data volume of repeatedly downloaded elements for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

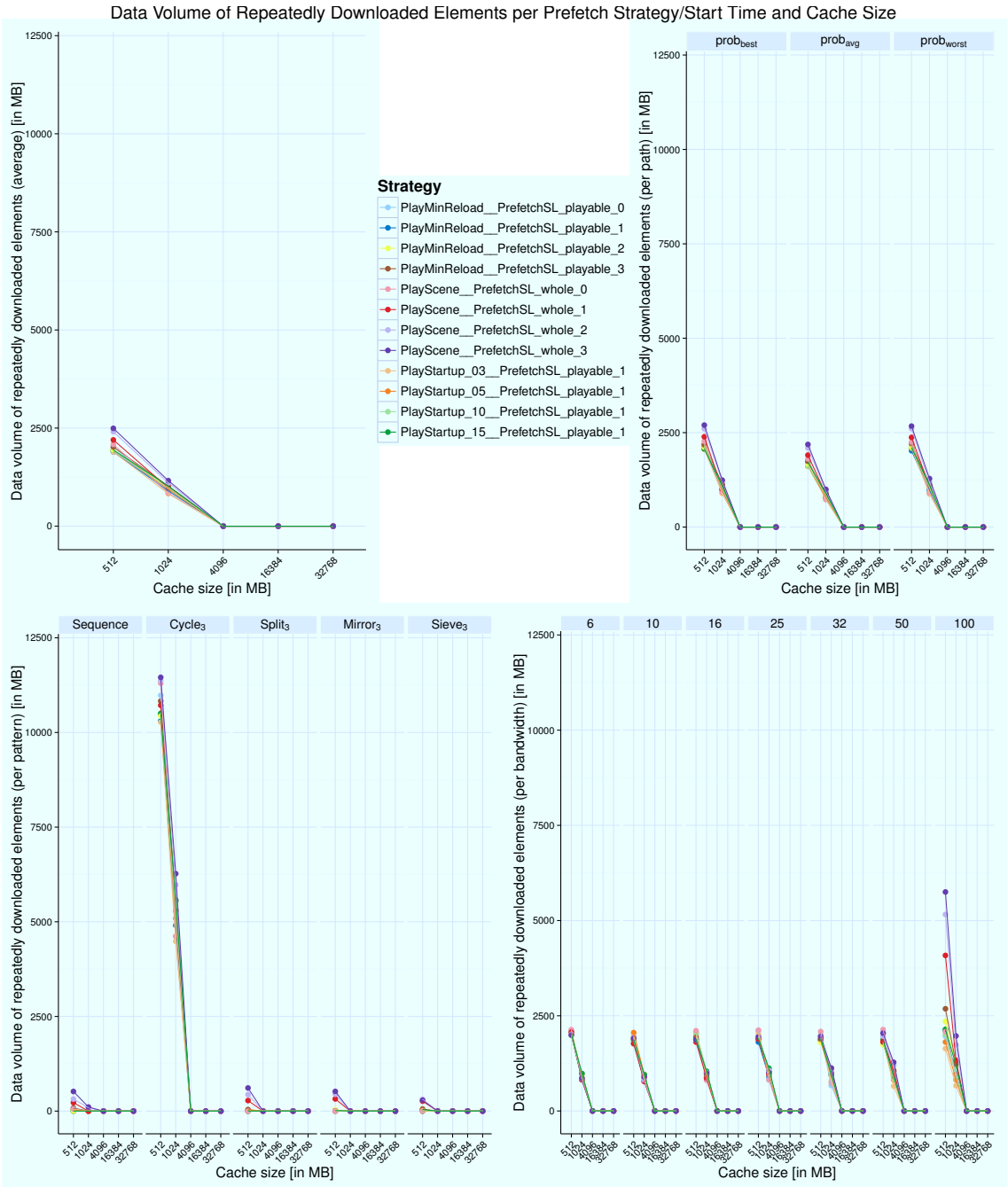


Figure I.13.: Evaluation of the pre-fetch strategies (detailed results) - data volume of repeatedly downloaded elements for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

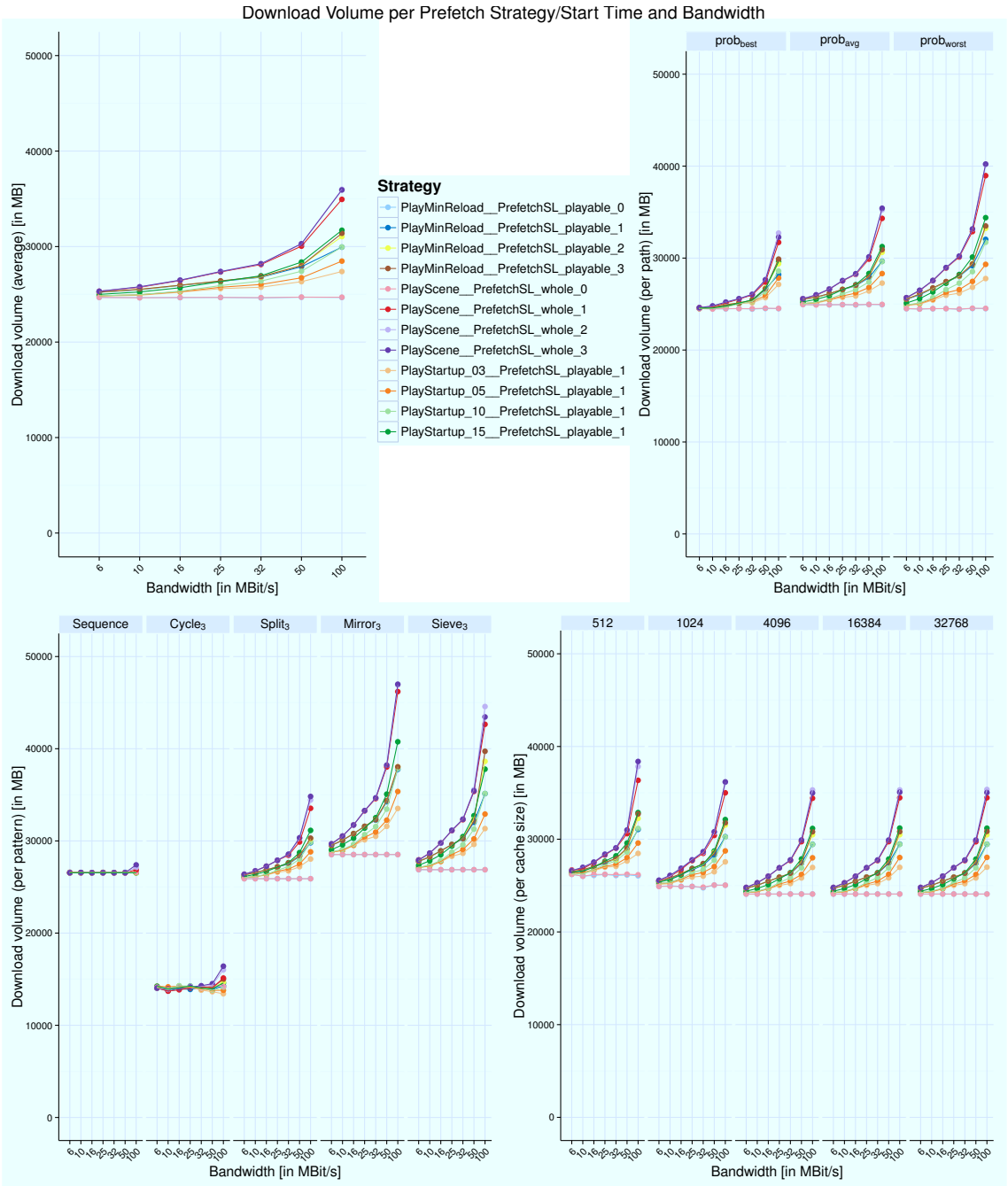


Figure I.14.: Evaluation of the pre-fetch strategies (detailed results) - download volume for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

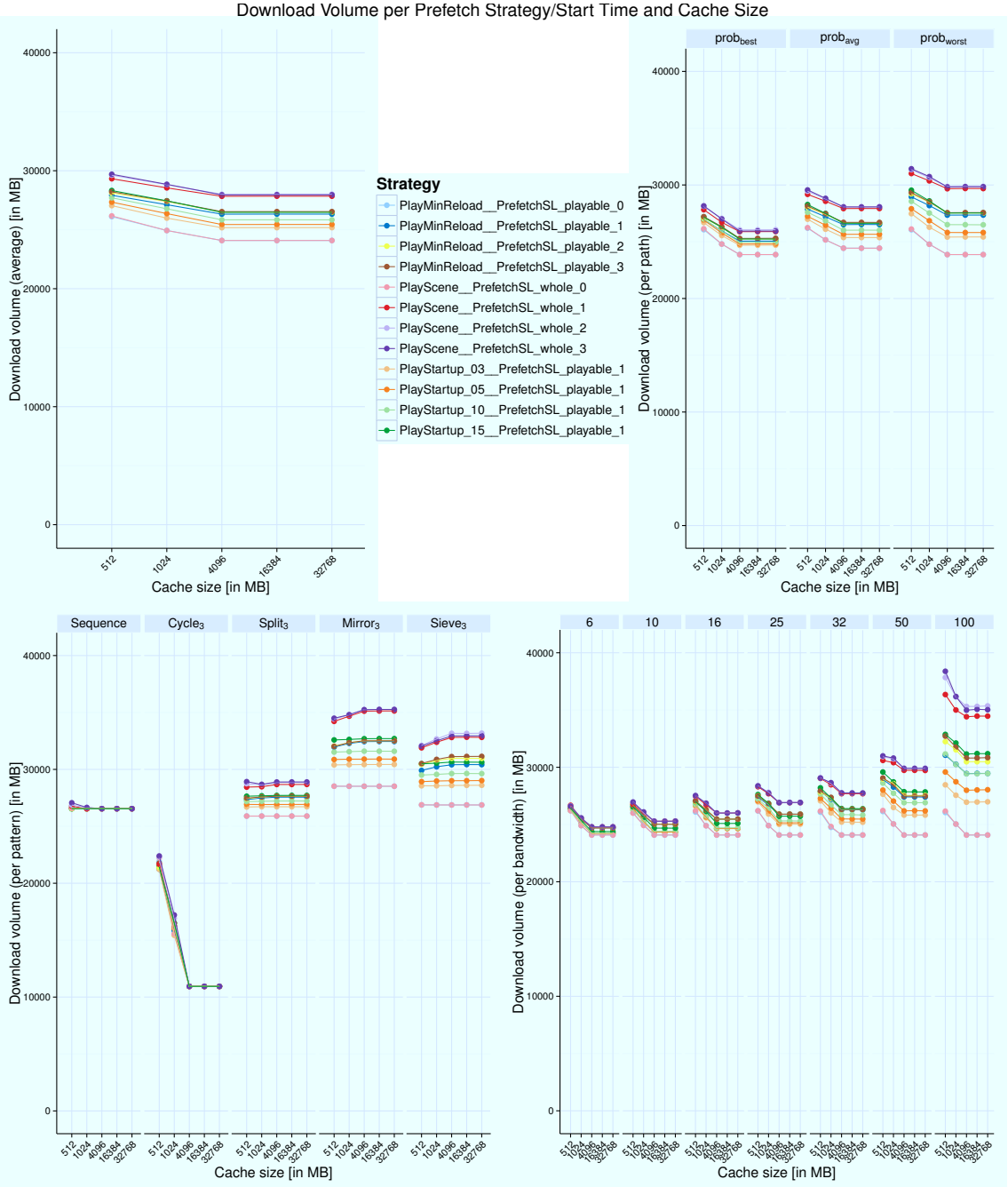


Figure I.15.: Evaluation of the pre-fetch strategies (detailed results) - download volume for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

I.3. Evaluation of the Delete Strategies

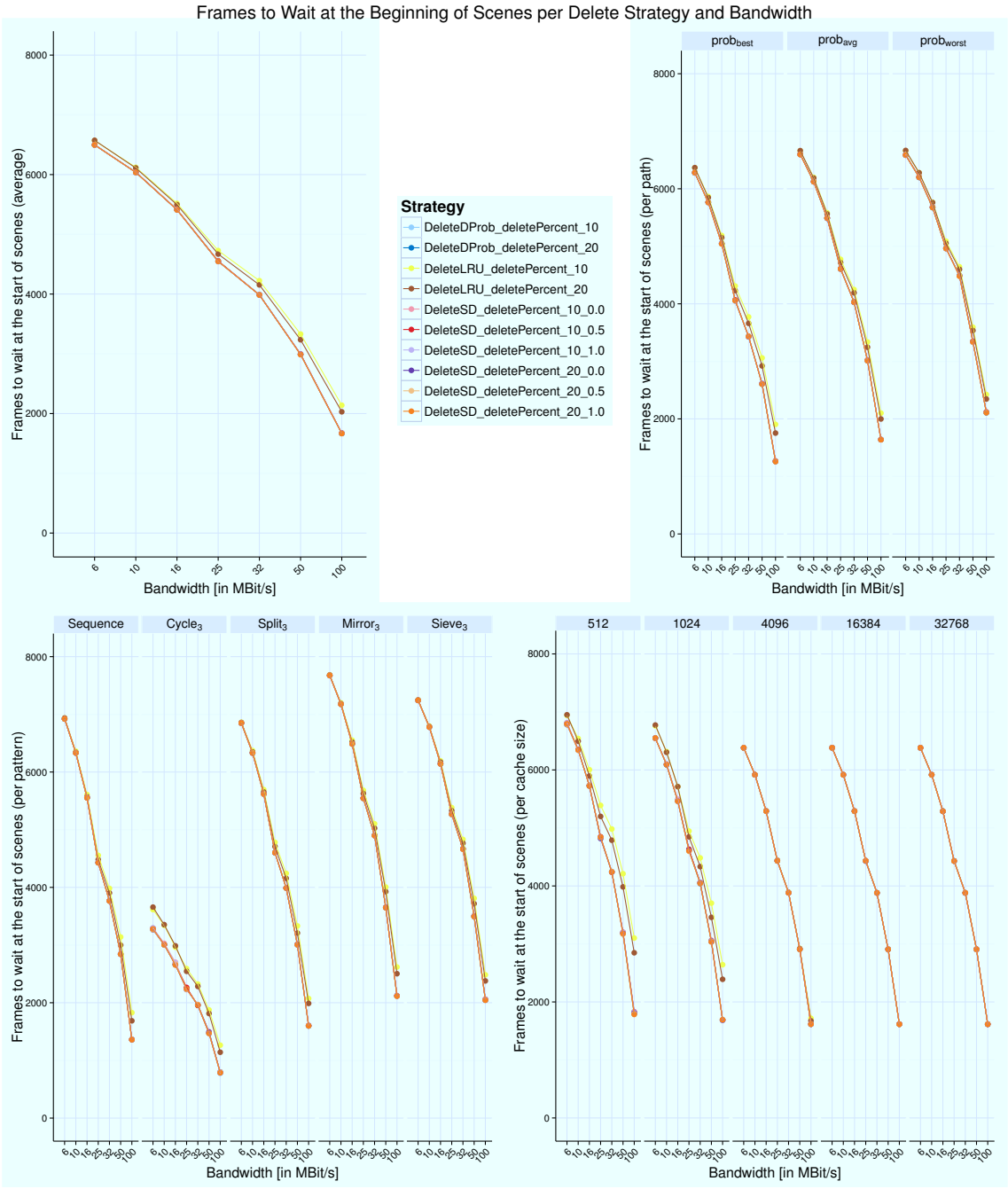


Figure I.16.: Evaluation of the delete strategies (detailed results) - frames to wait before playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

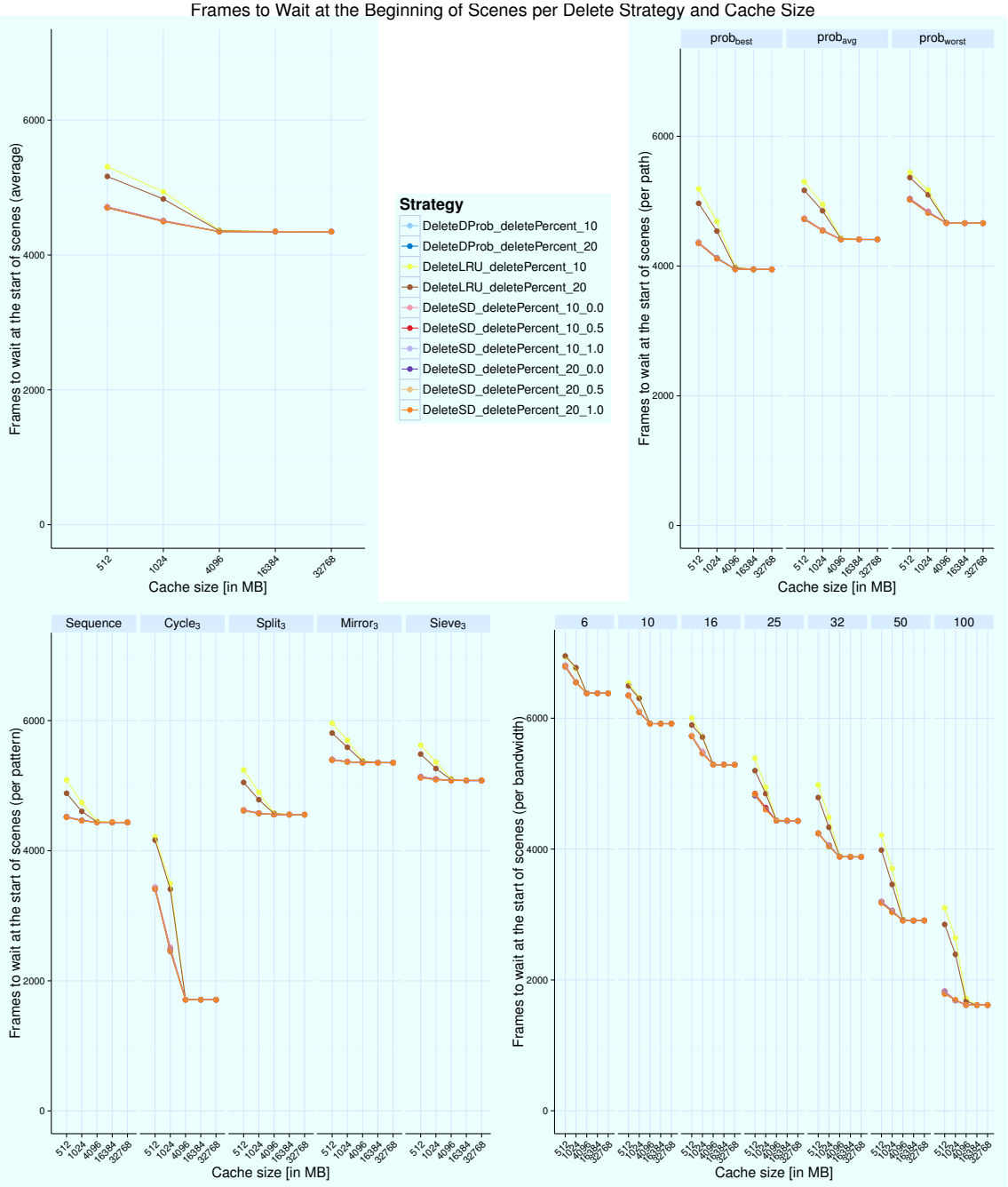


Figure I.17.: Evaluation of the delete strategies (detailed results) - frames to wait before playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

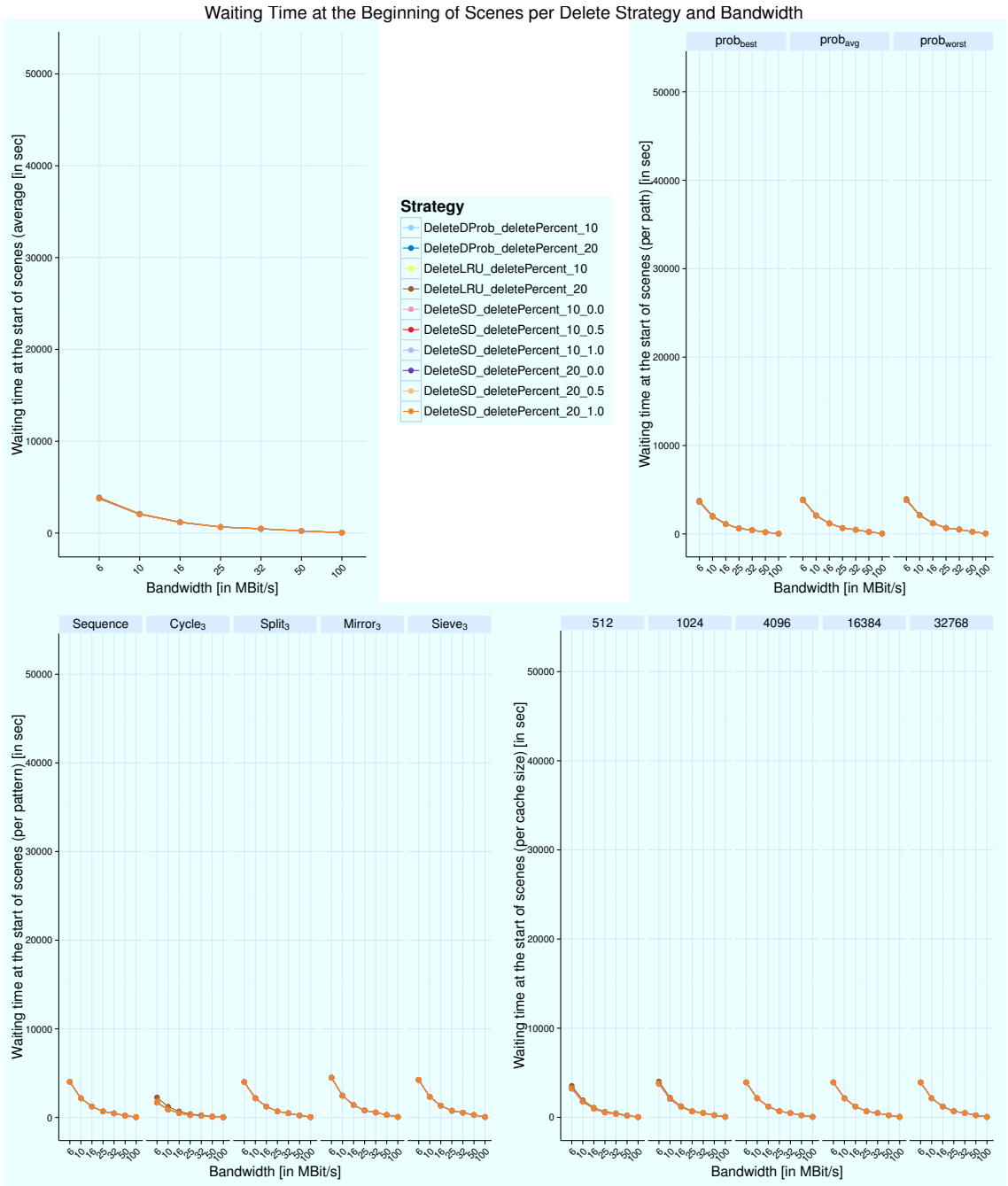


Figure I.18.: Evaluation of the delete strategies (detailed results) - waiting time before playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

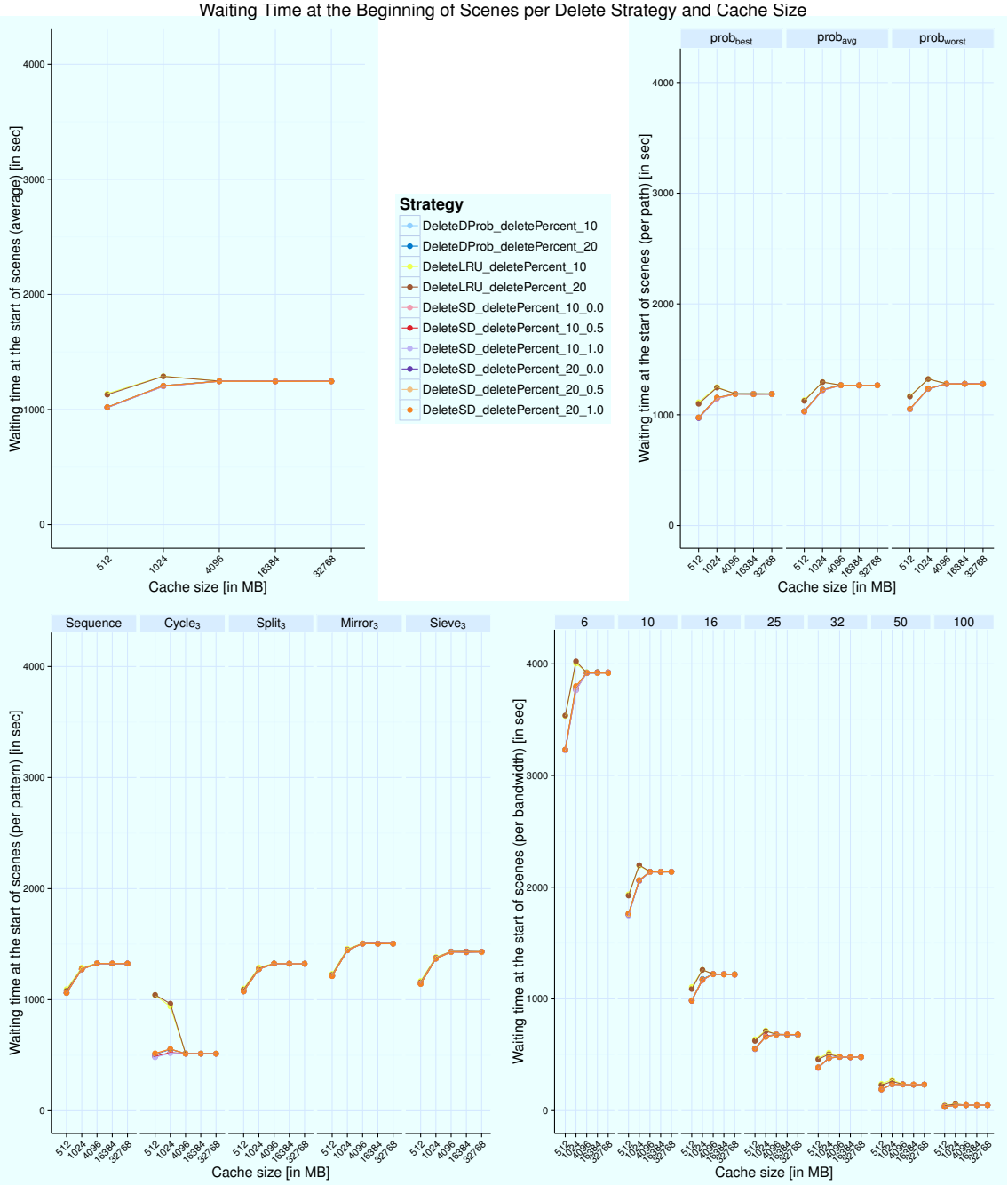


Figure I.19.: Evaluation of the delete strategies (detailed results) - waiting time before playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

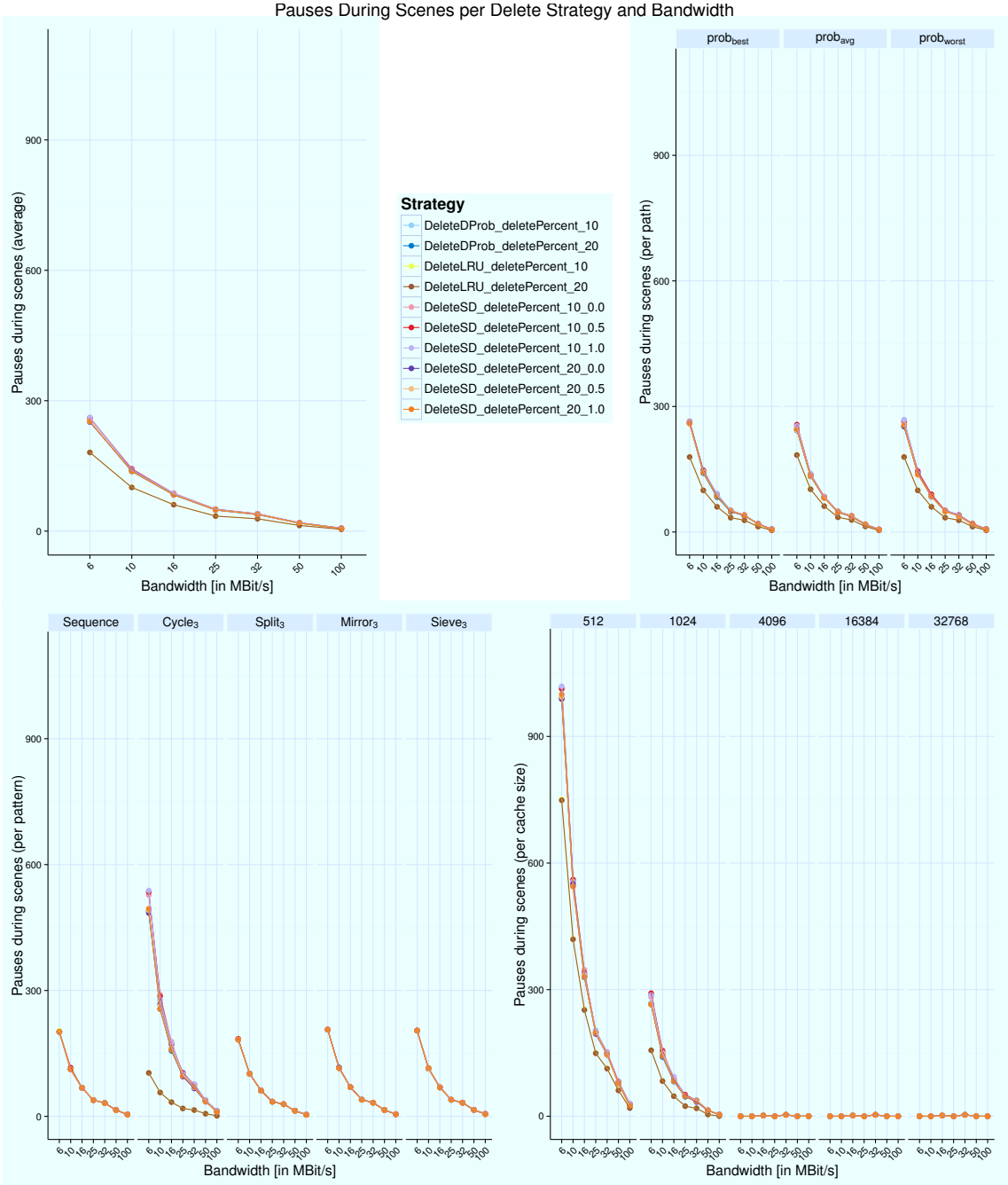


Figure I.20.: Evaluation of the delete strategies (detailed results) - pauses during playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

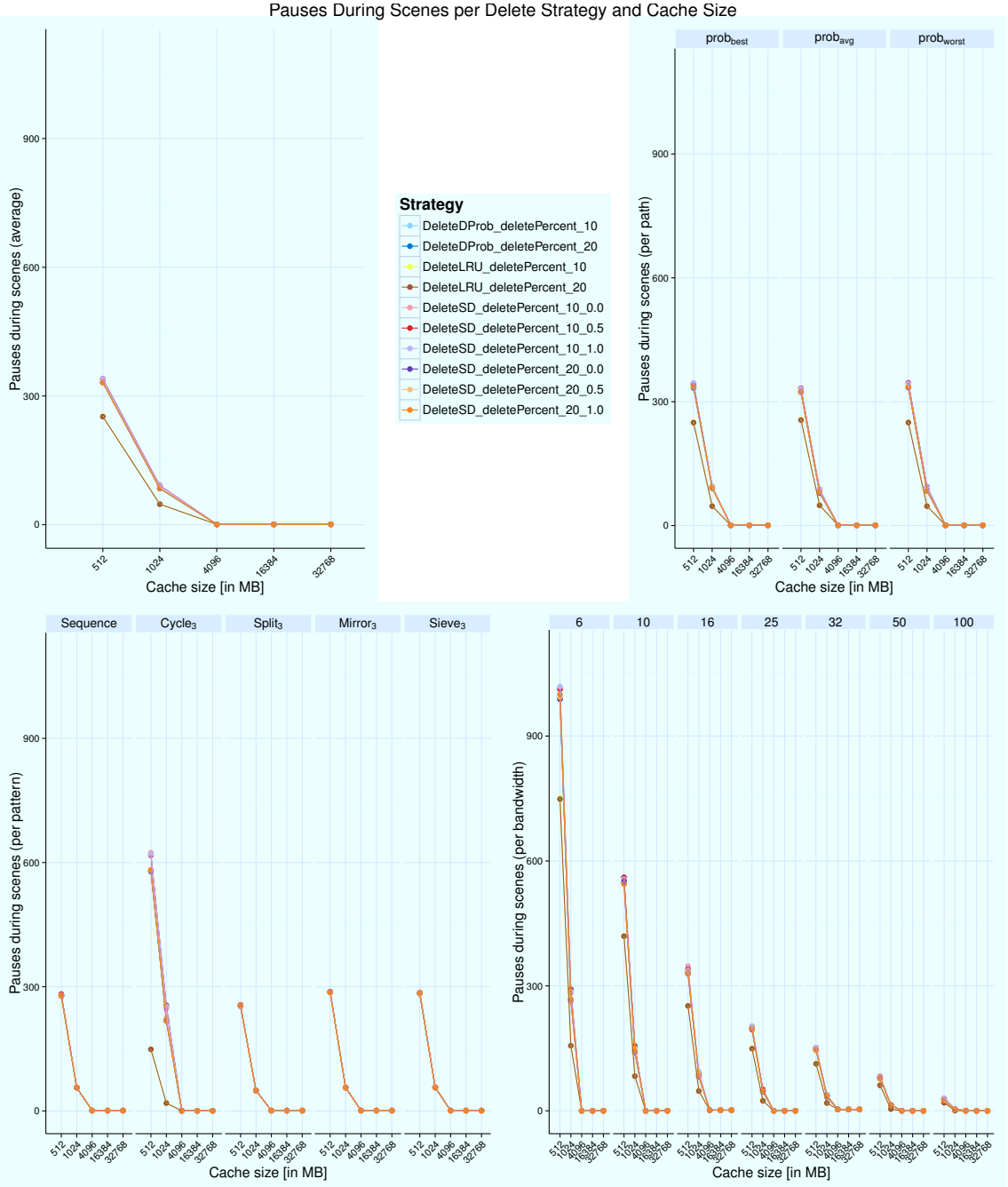


Figure I.21.: Evaluation of the delete strategies (detailed results) - pauses during playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

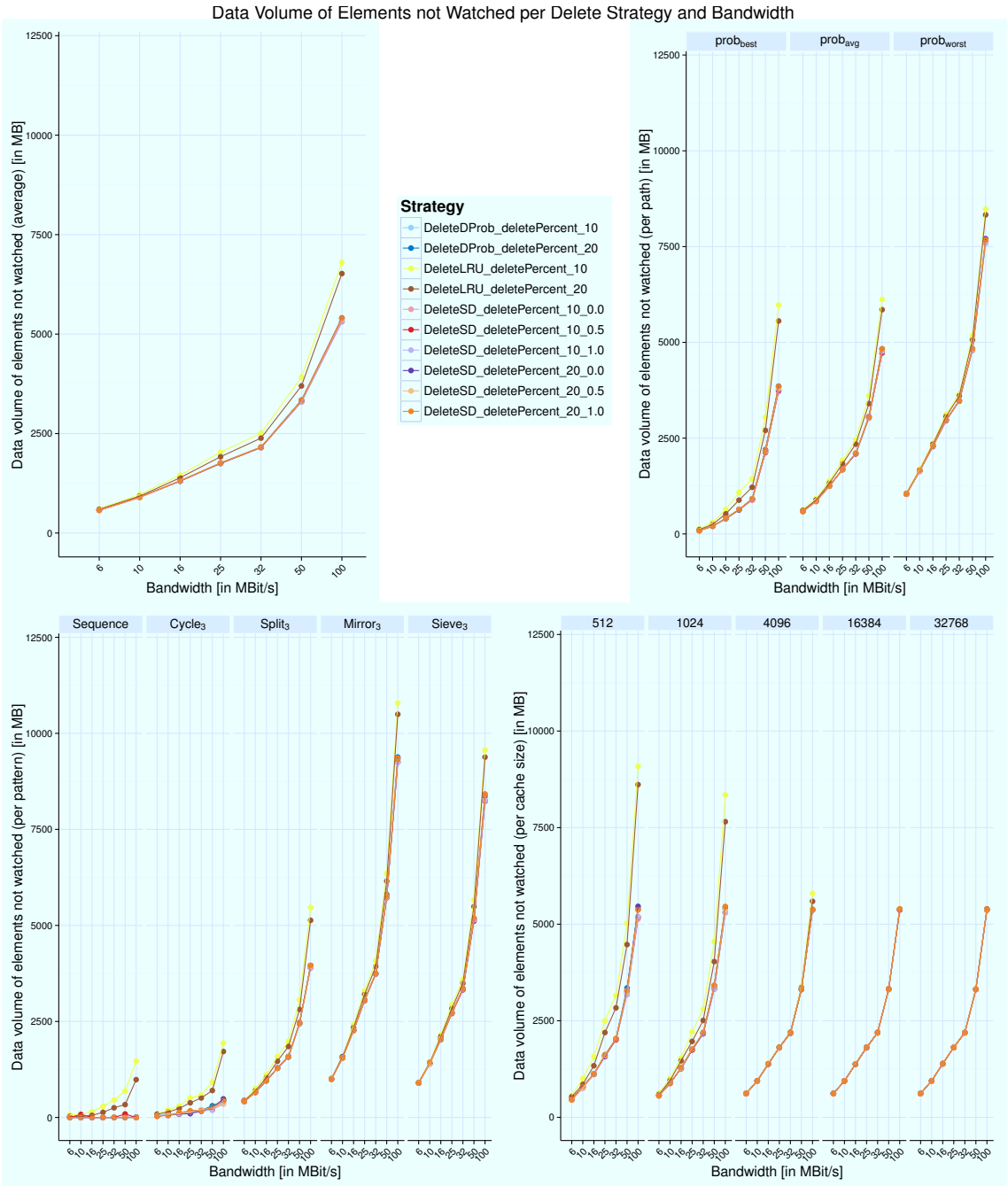


Figure I.22.: Evaluation of the delete strategies (detailed results) - data volume of elements not watched for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

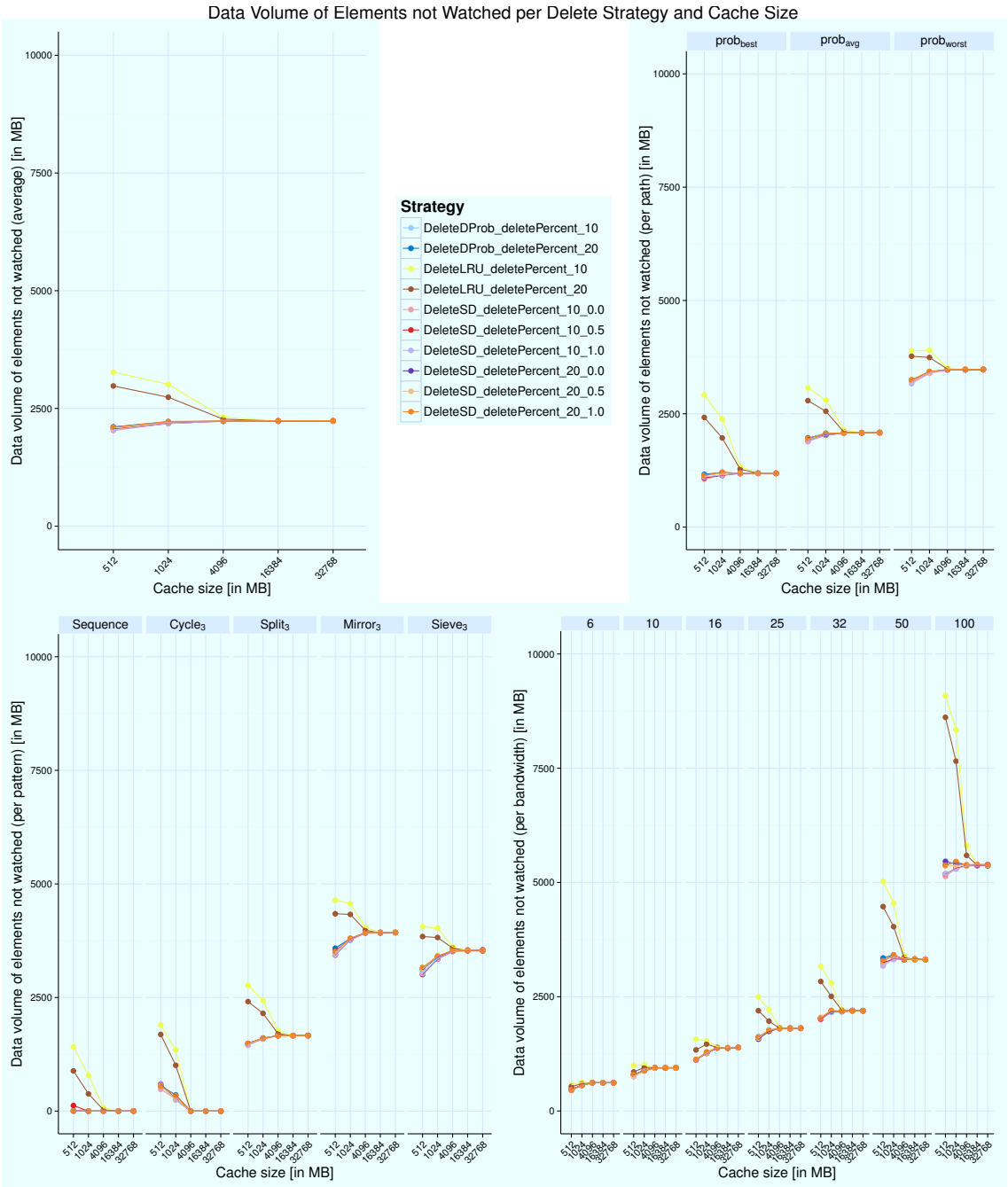


Figure I.23.: Evaluation of the delete strategies (detailed results) - data volume of elements not watched for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

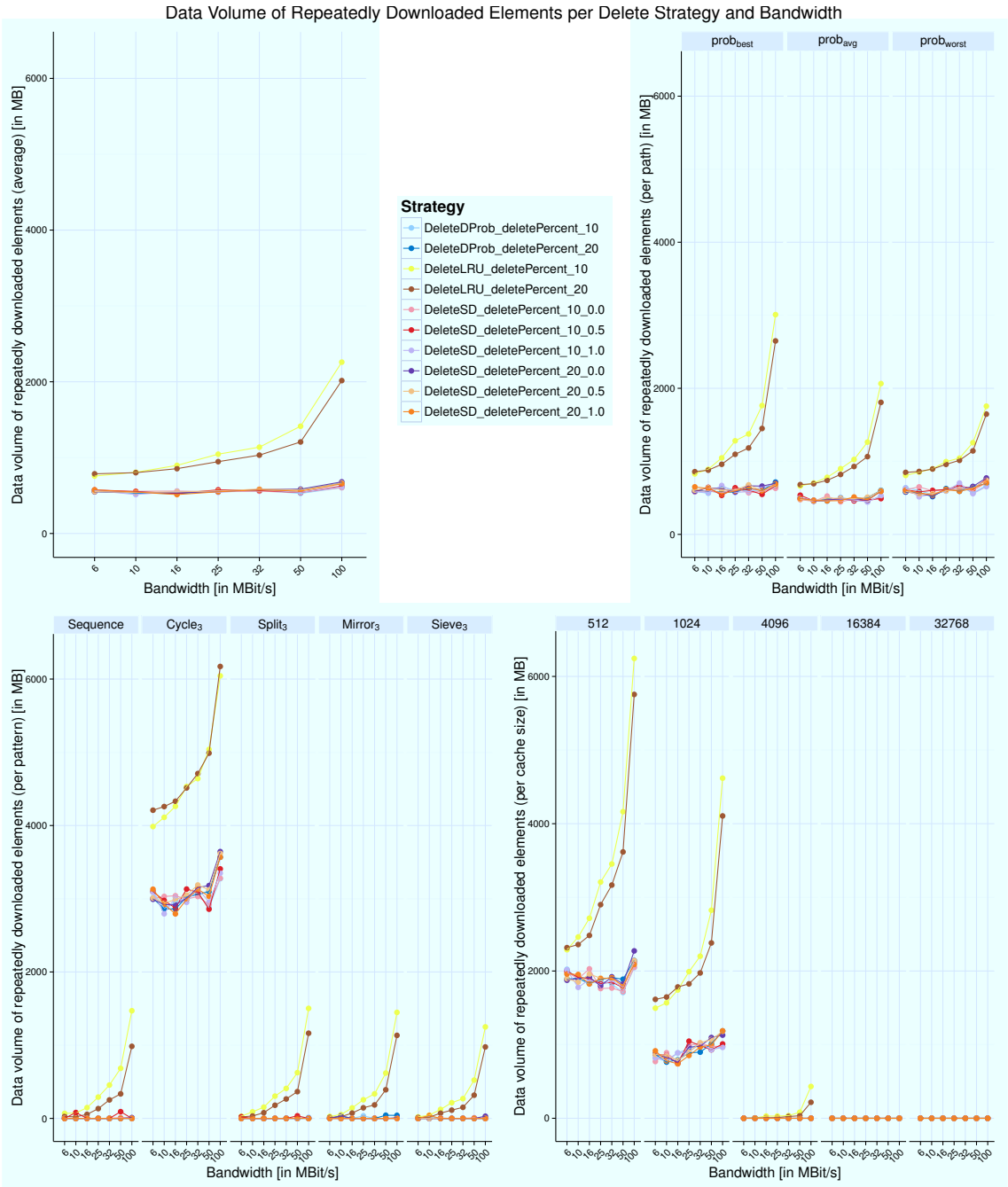


Figure I.24.: Evaluation of the delete strategies (detailed results) - data volume of repeatedly downloaded elements for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

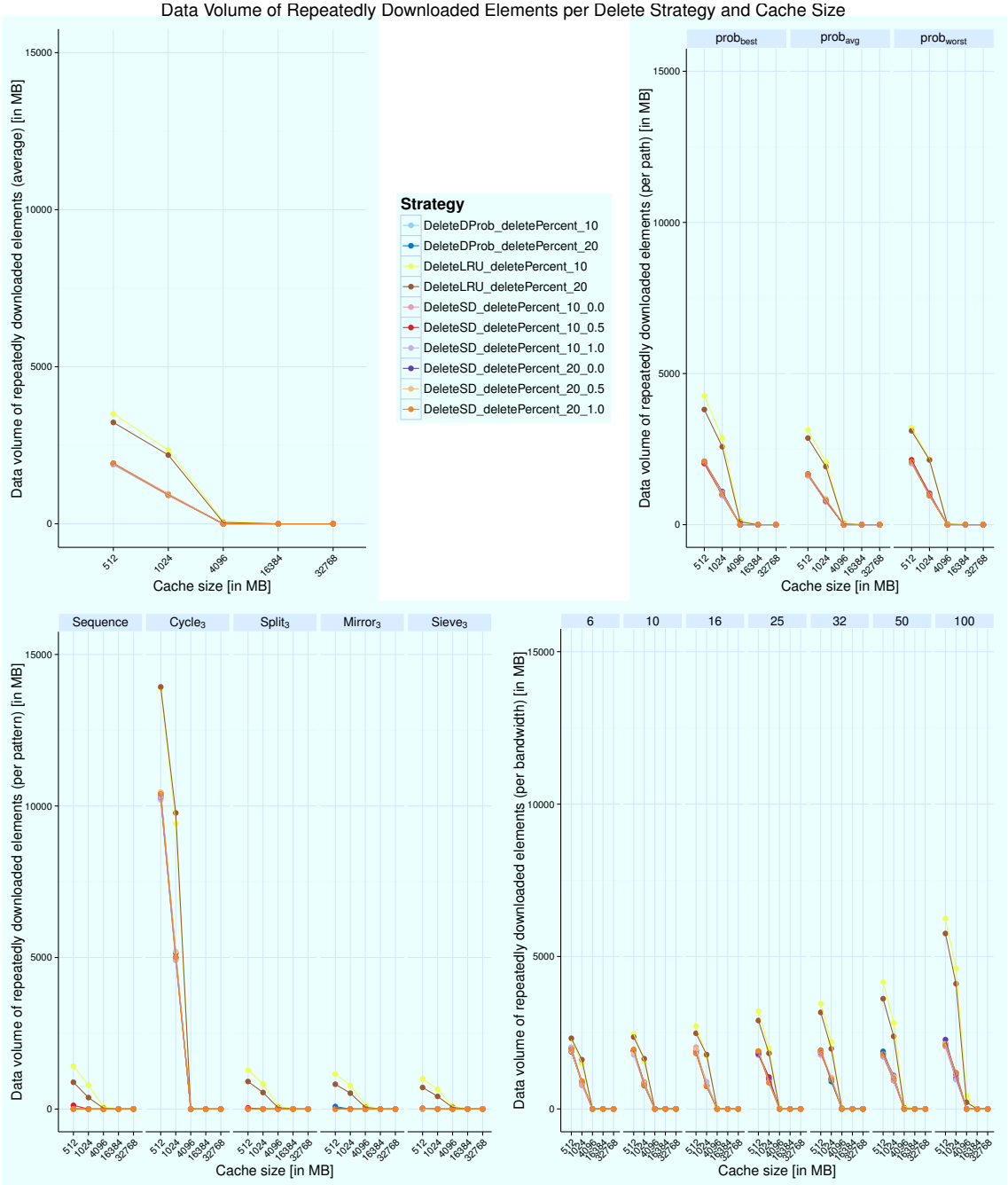


Figure I.25.: Evaluation of the delete strategies (detailed results) - data volume of repeatedly downloaded elements for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

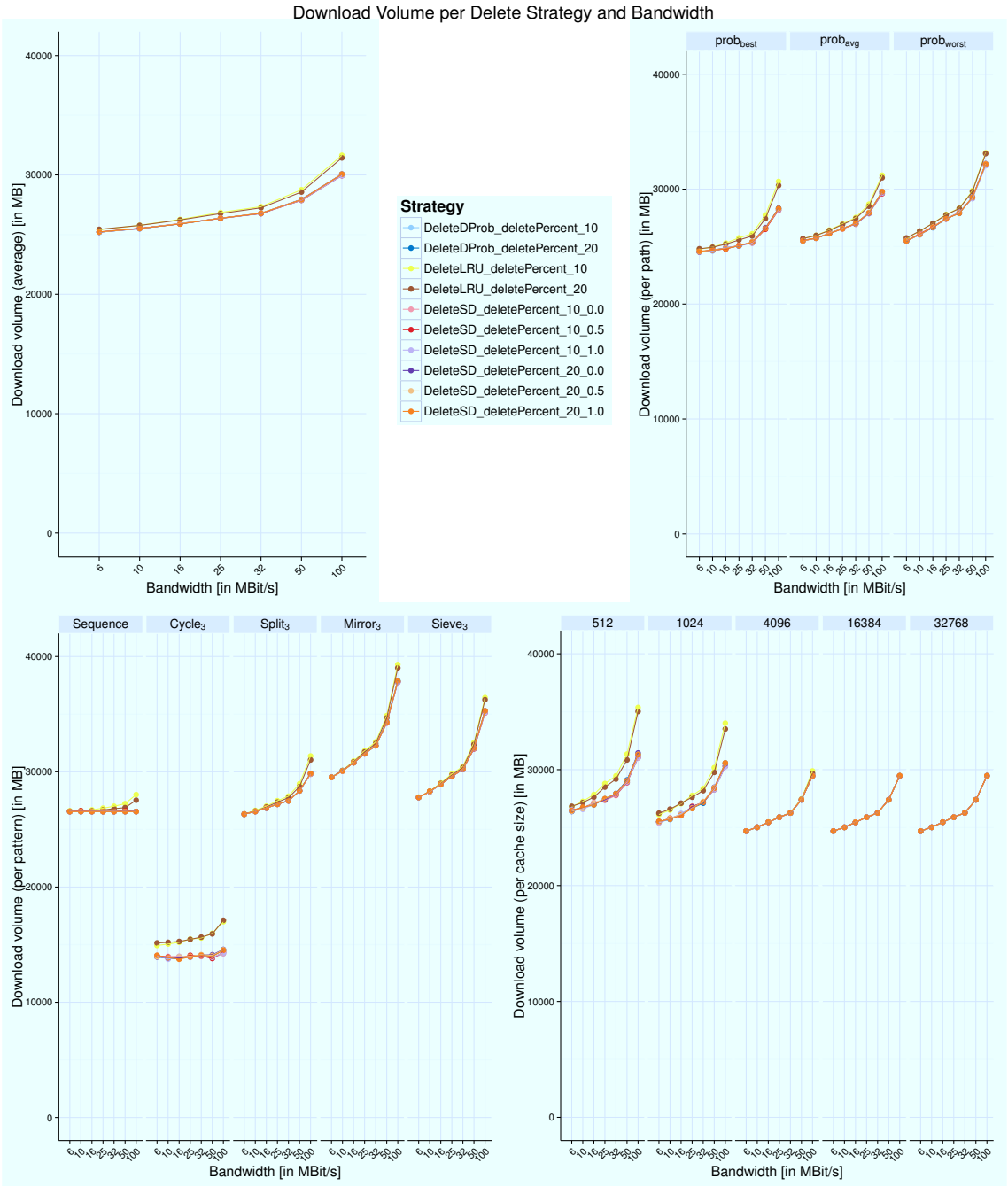


Figure I.26.: Evaluation of the delete strategies (detailed results) - download volume for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

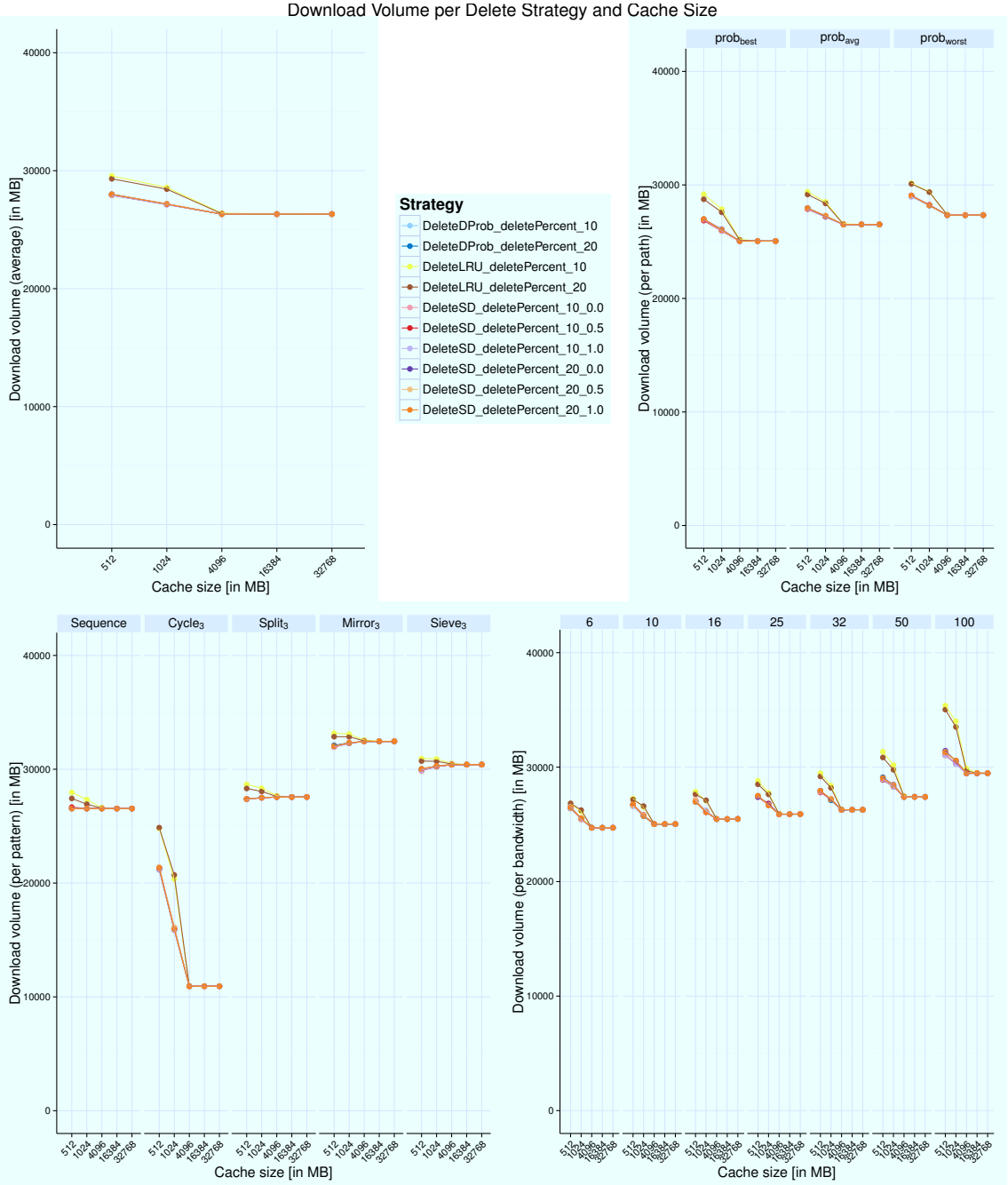


Figure I.27.: Evaluation of the delete strategies (detailed results) - download volume for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

I.4. Evaluation of the Number of Annotations

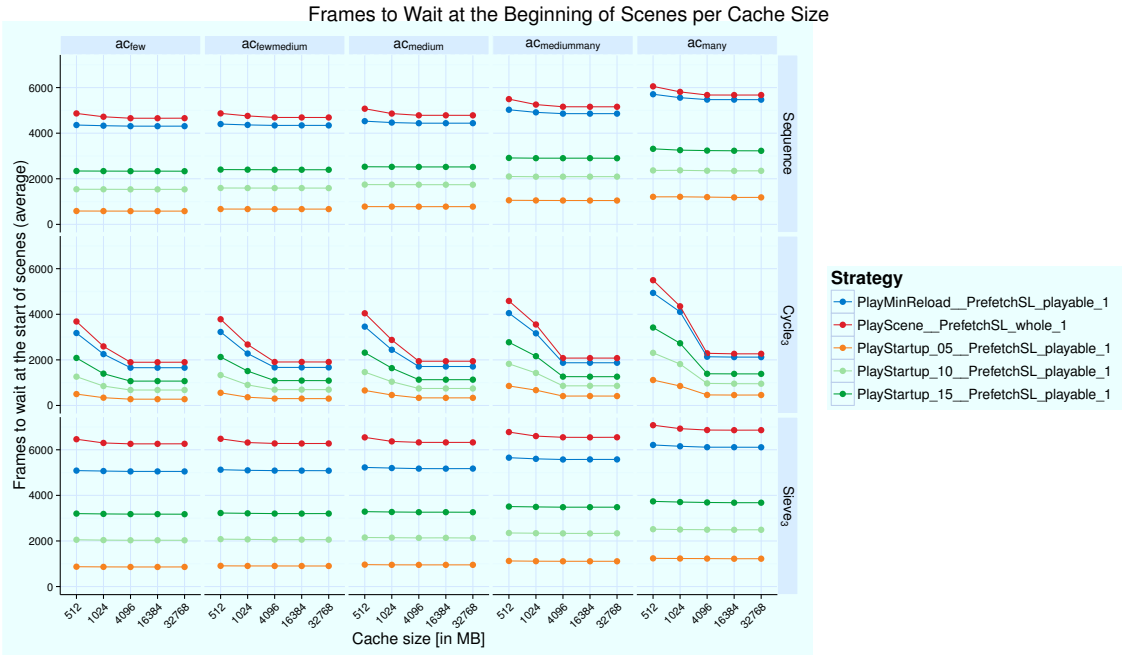


Figure I.28.: Evaluation of the number of annotations (selected strategies) - frames to wait before playback for different cache sizes: annotations × pattern.

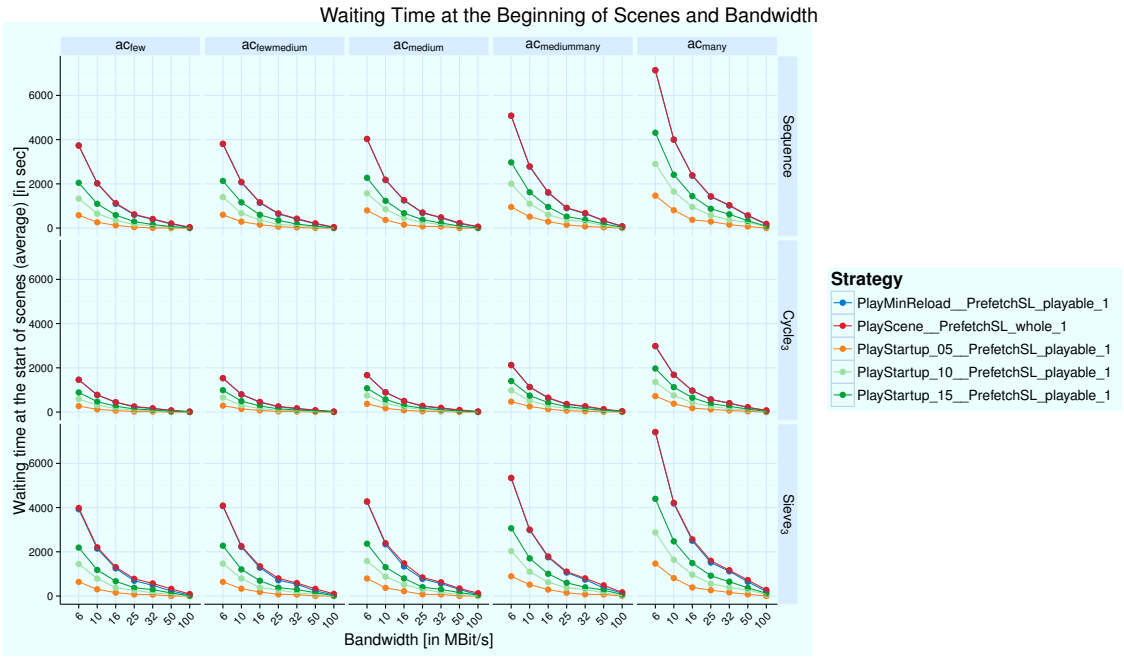


Figure I.29.: Evaluation of the number of annotations (selected strategies) - waiting time before playback for different bandwidths: annotations × pattern.

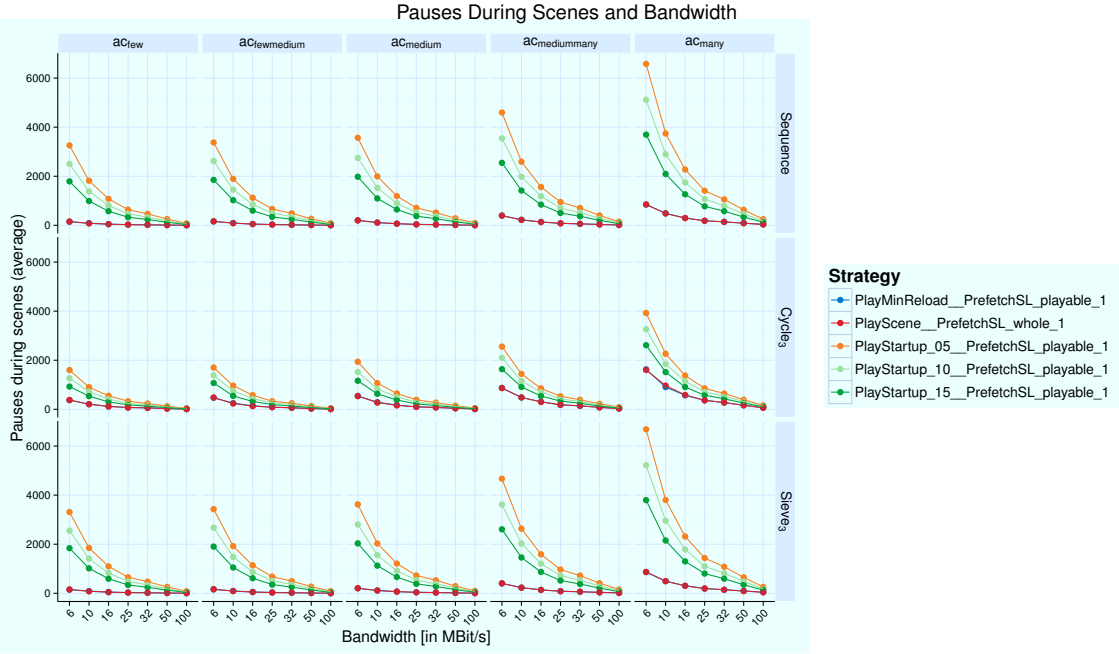


Figure I.30.: Evaluation of the number of annotations (selected strategies) - pauses during scenes for different bandwidths: annotations × pattern.

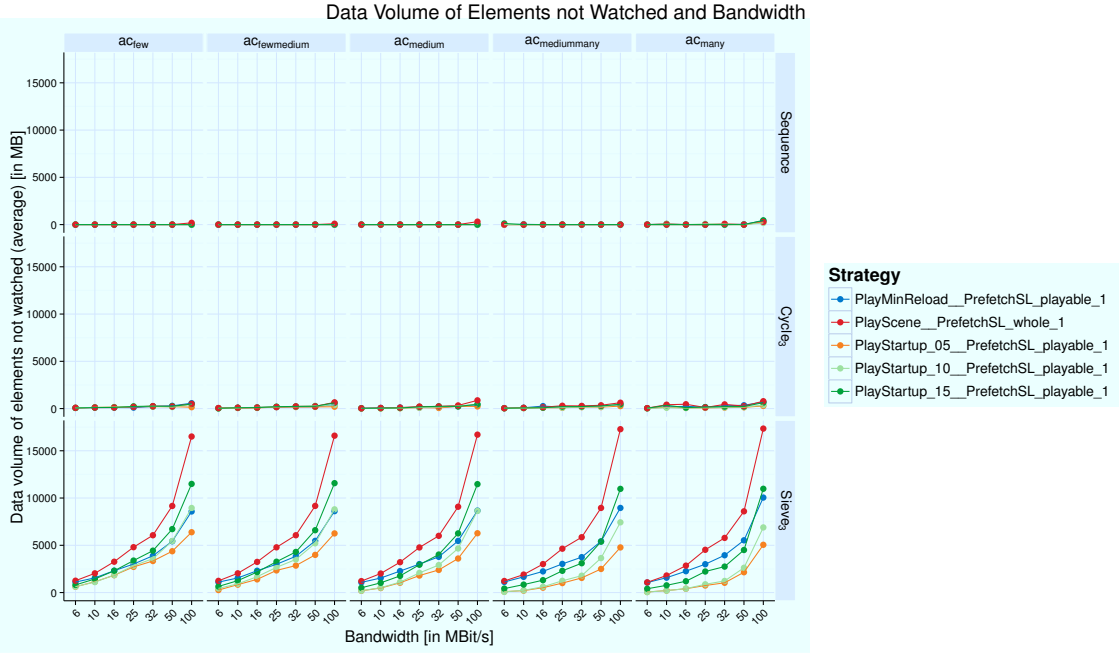


Figure I.31.: Evaluation of the number of annotations (selected strategies) - data volume of elements not watched for different bandwidths: number of annotations × pattern.

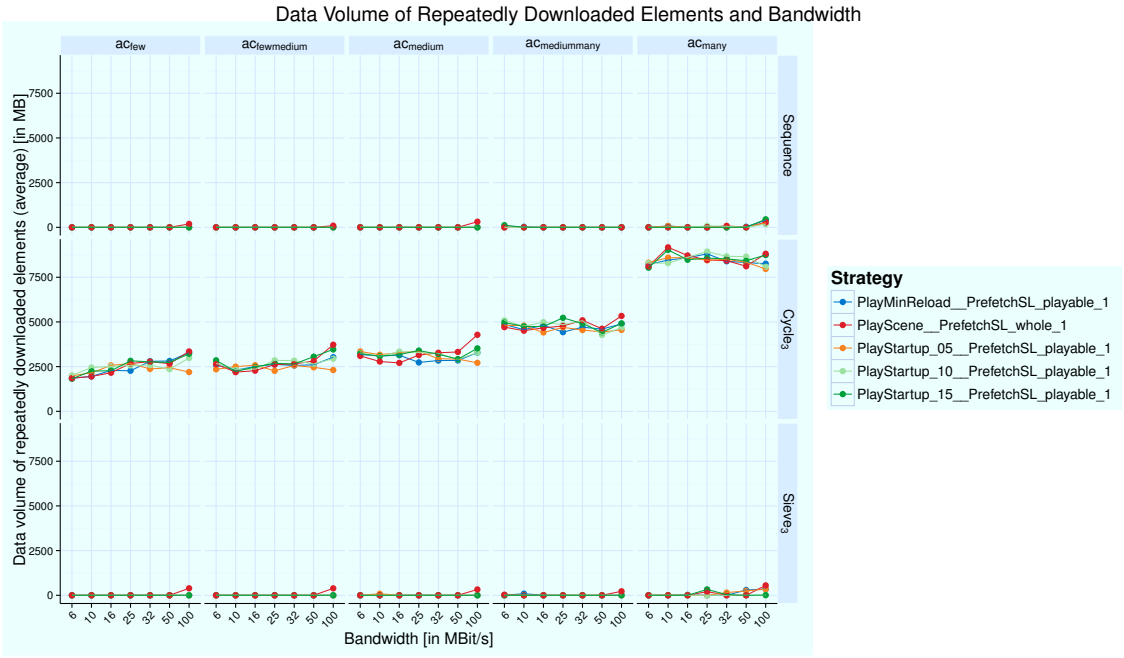


Figure I.32.: Evaluation of the number of annotations (selected strategies) - data volume of repeatedly downloaded elements for different bandwidths: number of annotations × pattern.

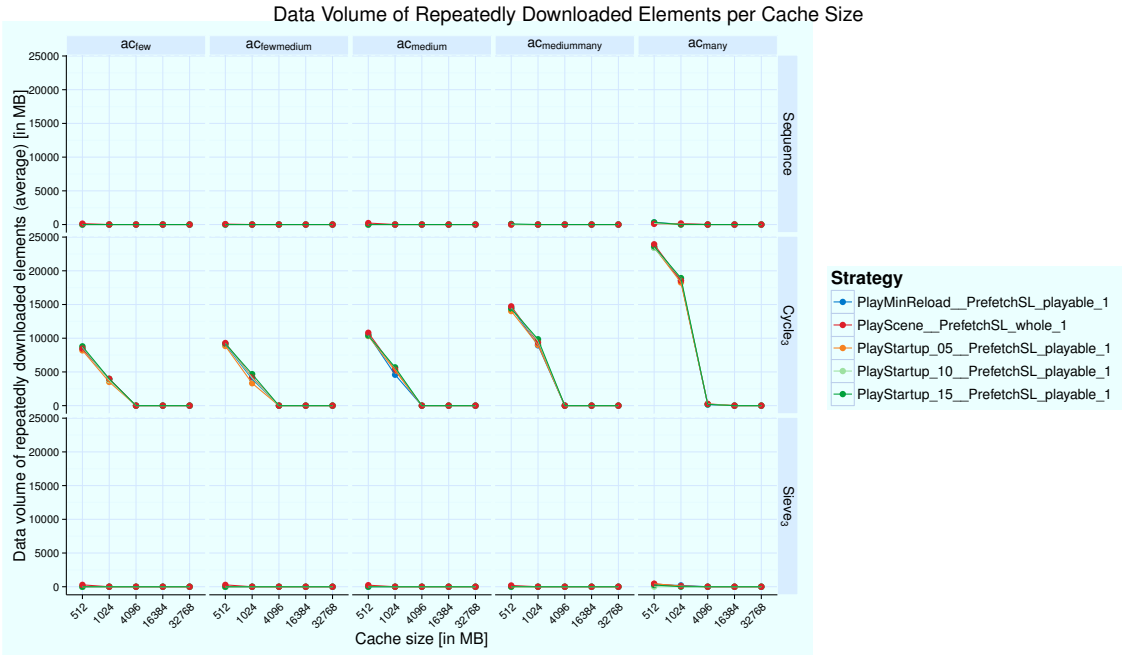


Figure I.33.: Evaluation of the number of annotations (selected strategies) - data volume of repeatedly downloaded elements for different cache sizes: number of annotations × pattern.

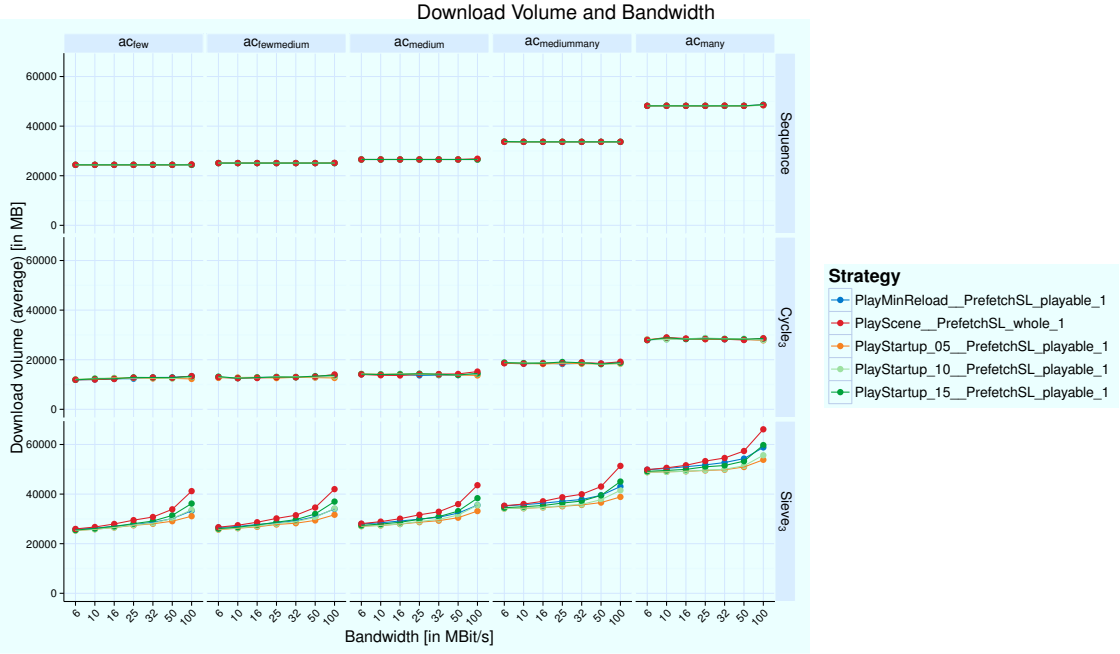


Figure I.34.: Evaluation of the number of annotations (selected strategies) - download volume the whole video for different bandwidths: number of annotations × pattern.

I.5. Evaluation of the Varying Path Probabilities and Pattern Widths

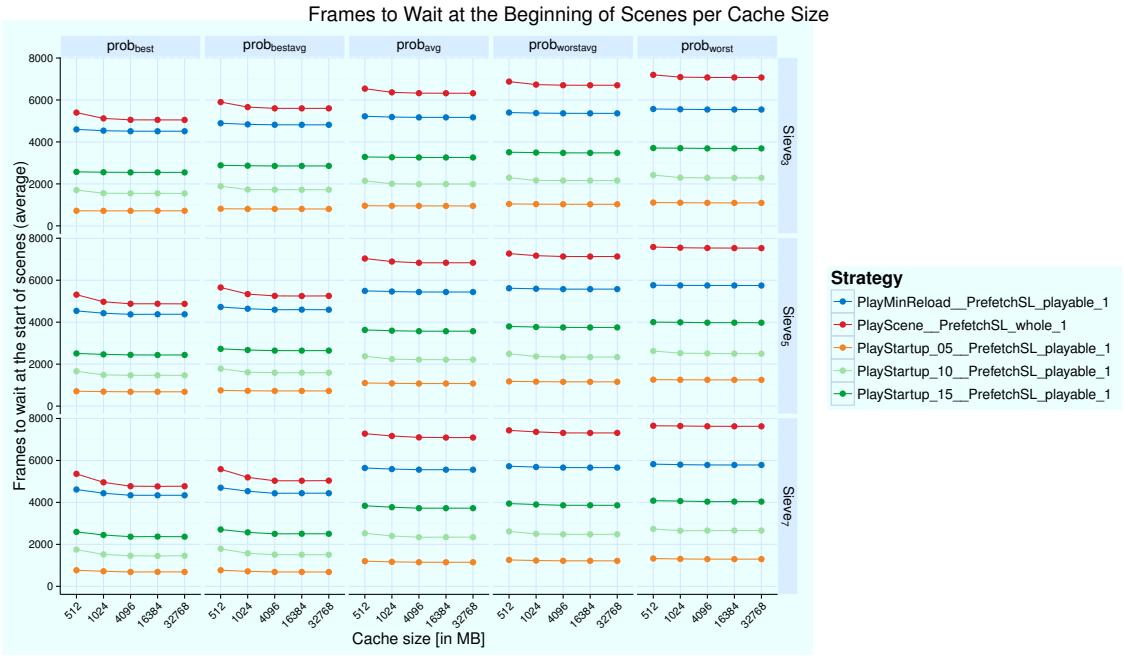


Figure I.35.: Evaluation of the pattern width and probabilities (selected strategies) - frames to wait before playback for different cache sizes: pattern width × probability.

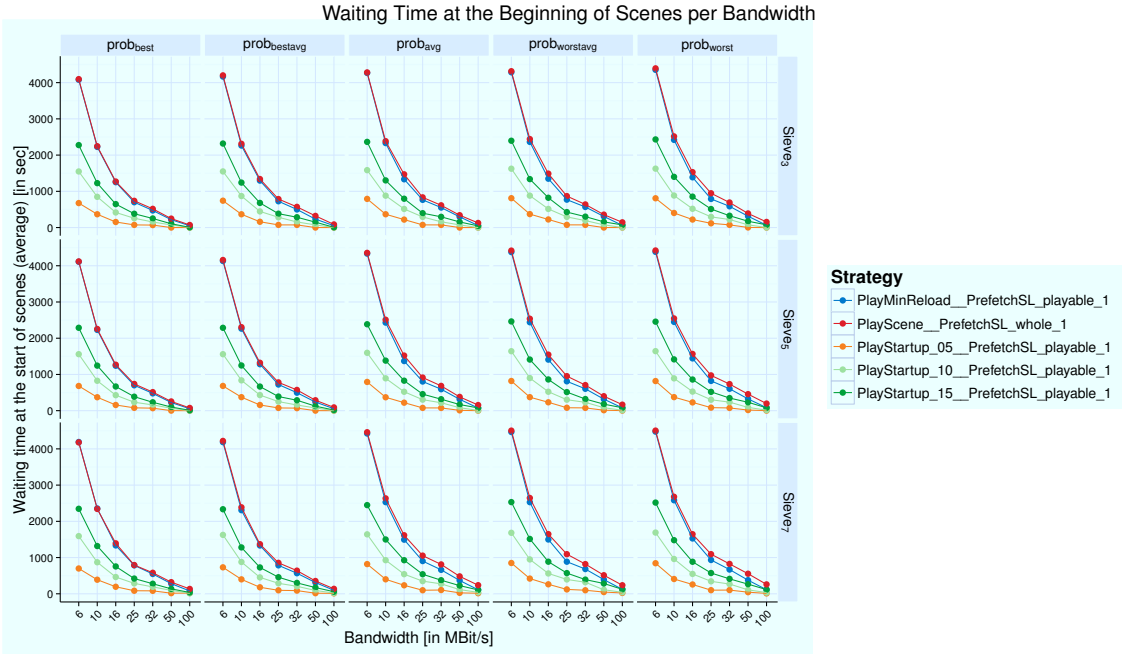


Figure I.36.: Evaluation of the pattern width and probabilities (selected strategies) - waiting time before playback for different bandwidths: pattern width × probability.

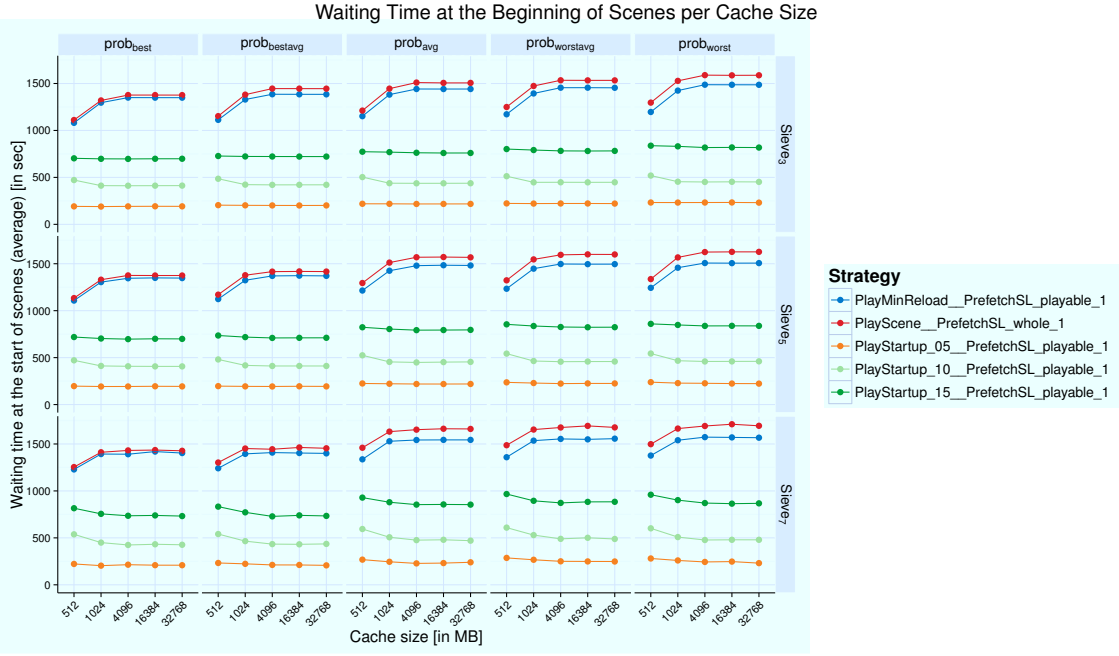


Figure I.37.: Evaluation of the pattern width and probabilities (selected strategies) - waiting time before playback for different cache sizes: pattern width × probability.

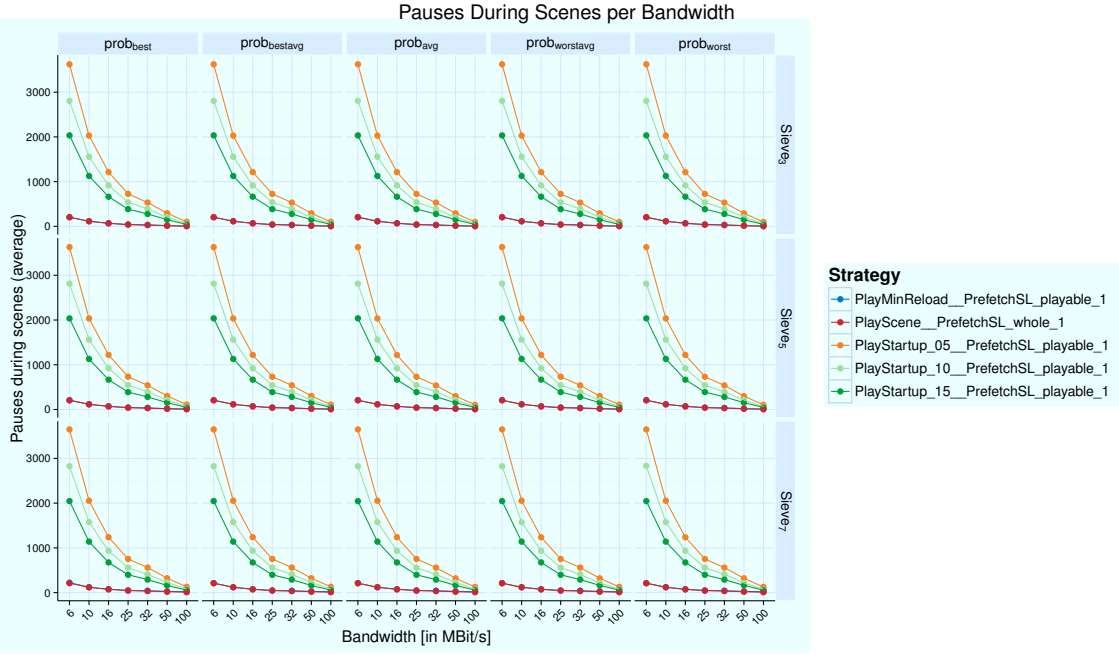


Figure I.38.: Evaluation of the pattern width and probabilities (selected strategies) - pauses during scenes for different bandwidths: pattern width × probability.

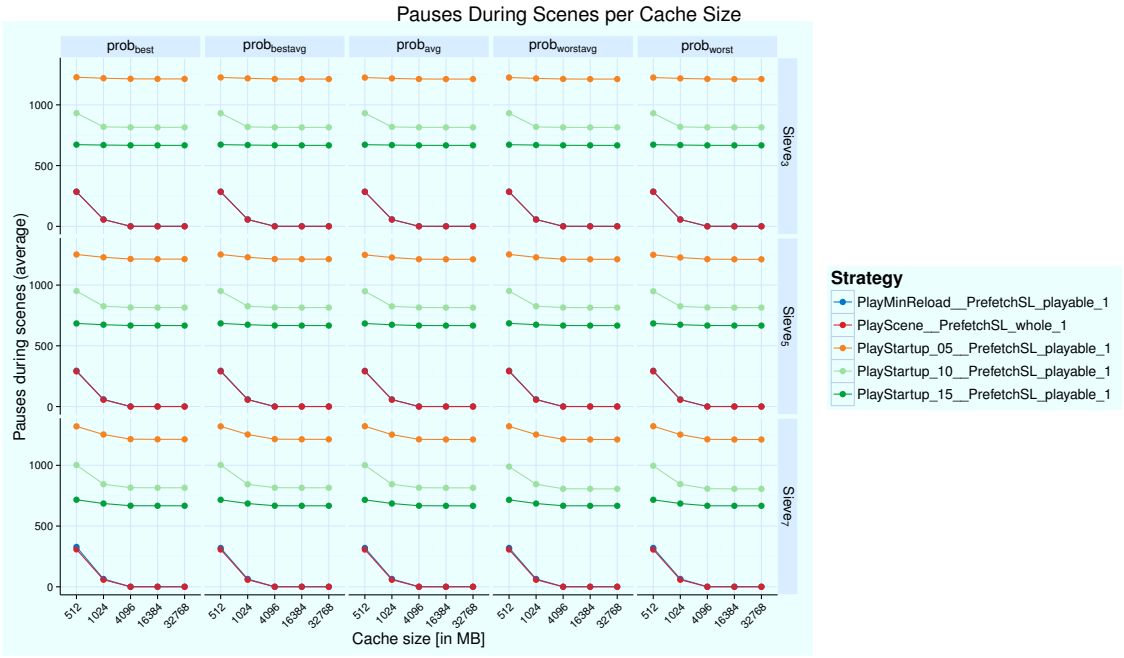


Figure I.39.: Evaluation of the pattern width and probabilities (selected strategies) - pauses during scenes for different cache sizes: pattern width × probability.

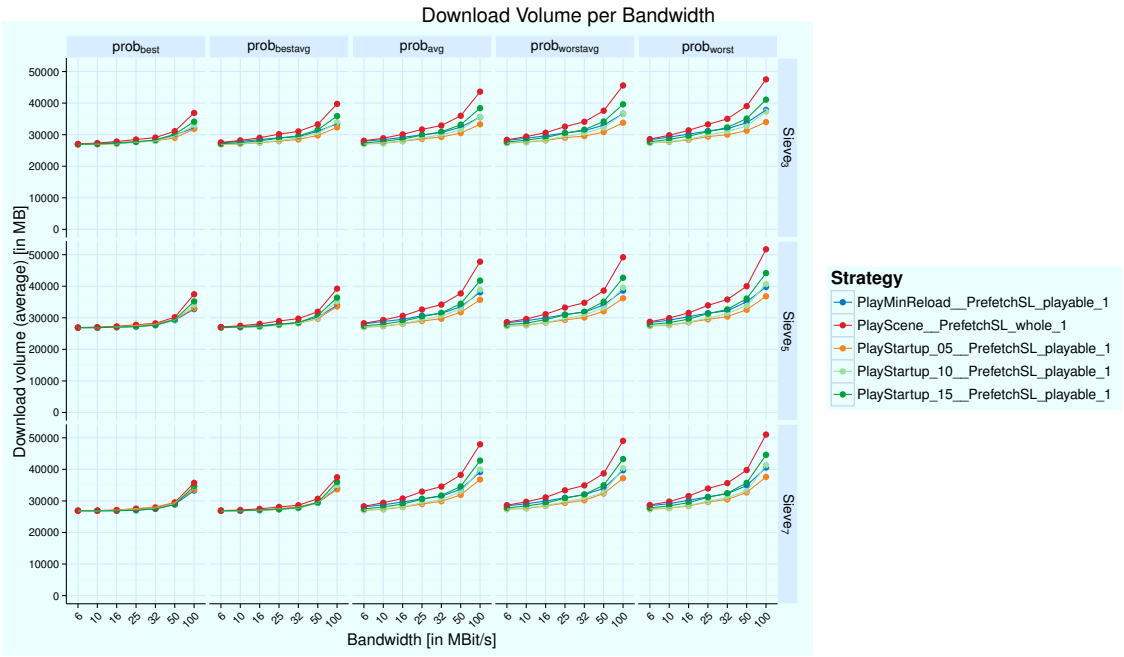


Figure I.40.: Evaluation of the pattern width and probabilities (selected strategies) - download volume the whole video for different bandwidths: pattern width × probability.

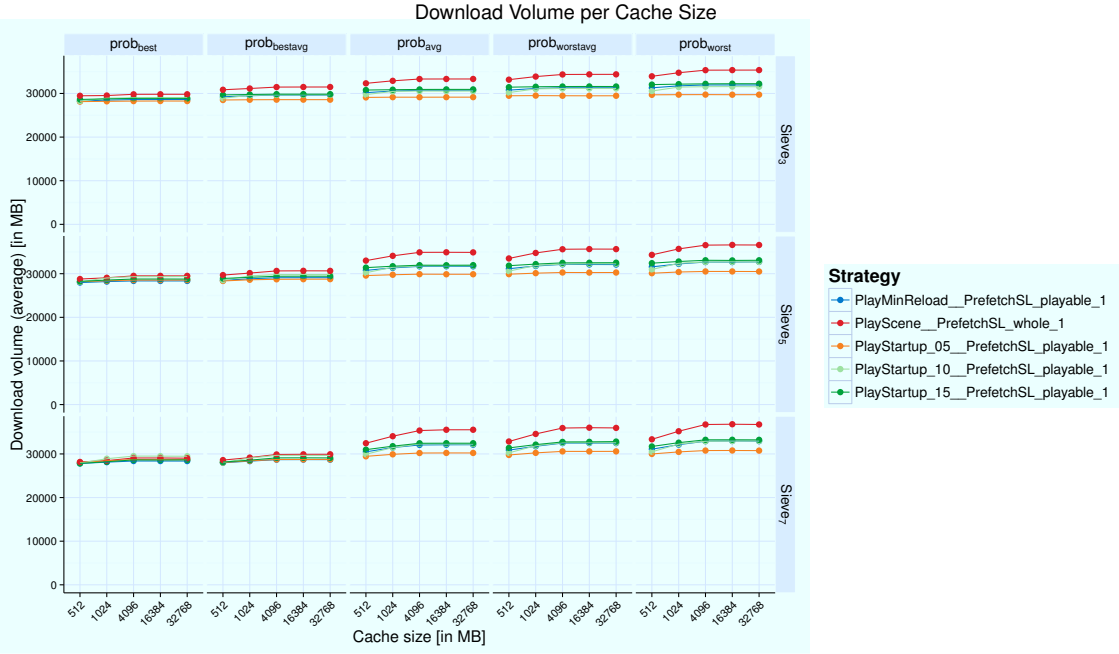


Figure I.41.: Evaluation of the pattern width and probabilities (selected strategies) - download volume the whole video for different cache sizes: pattern width × probability.

I.6. Evaluation of the Scenarios

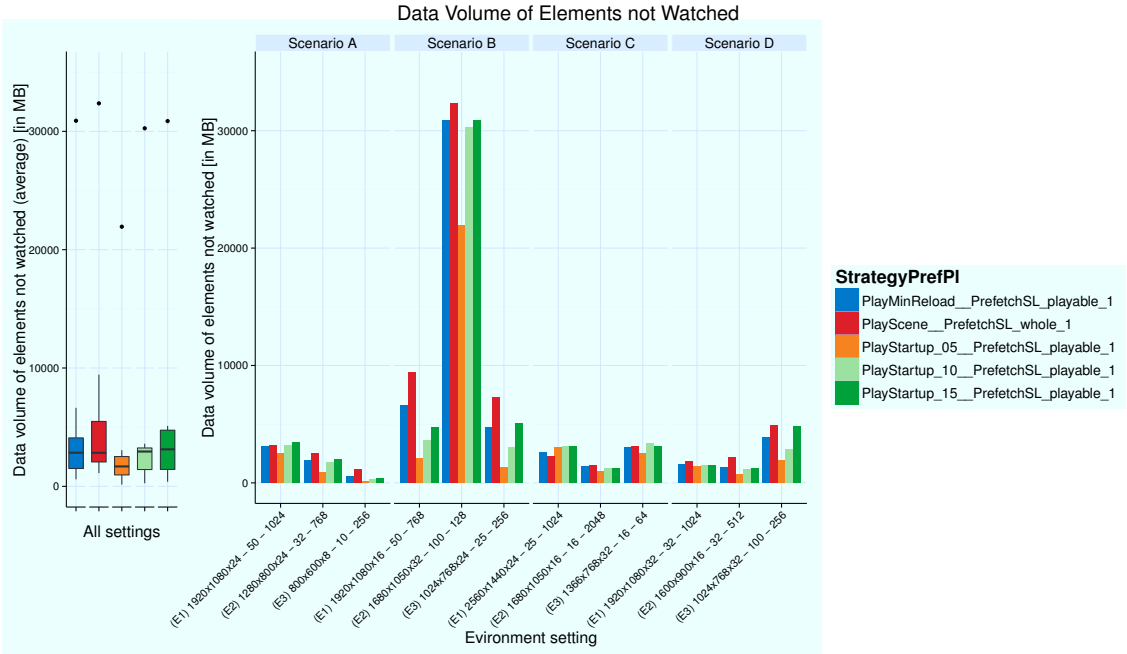


Figure I.42.: Evaluation of the scenarios - data volume of elements not watched: average for the whole test (left) and results grouped by scenario (right).

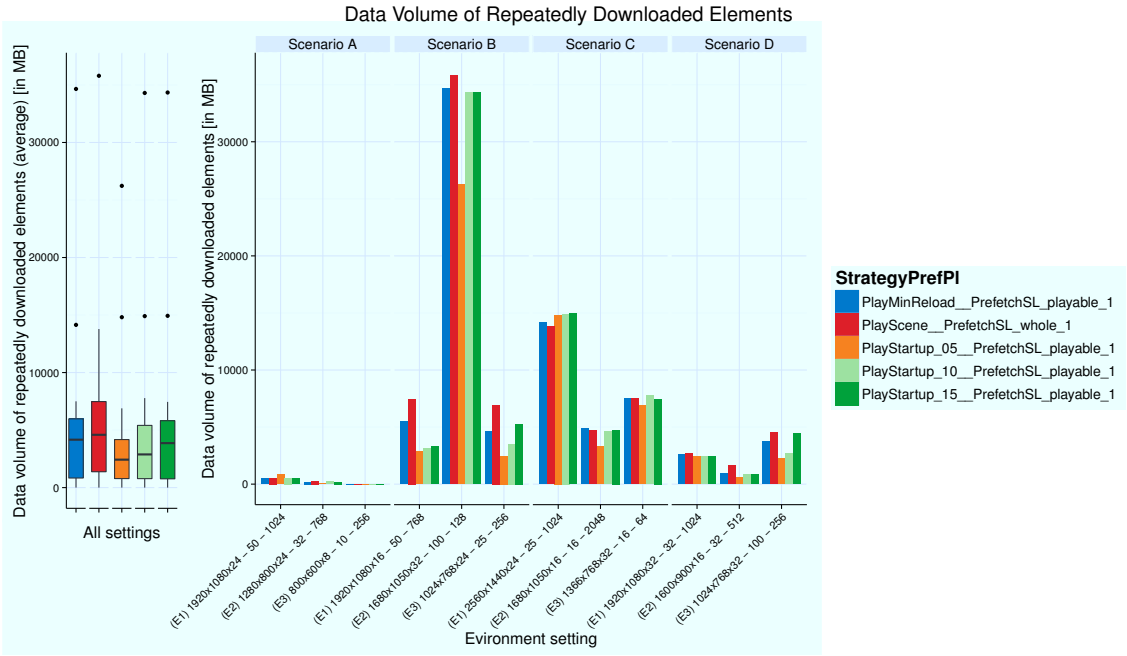


Figure I.43.: Evaluation of the scenarios - data volume of repeatedly downloaded elements: average for the whole test (left) and results grouped by scenario (right).

I.7. Evaluation of the Priority-based Algorithms/Strategies

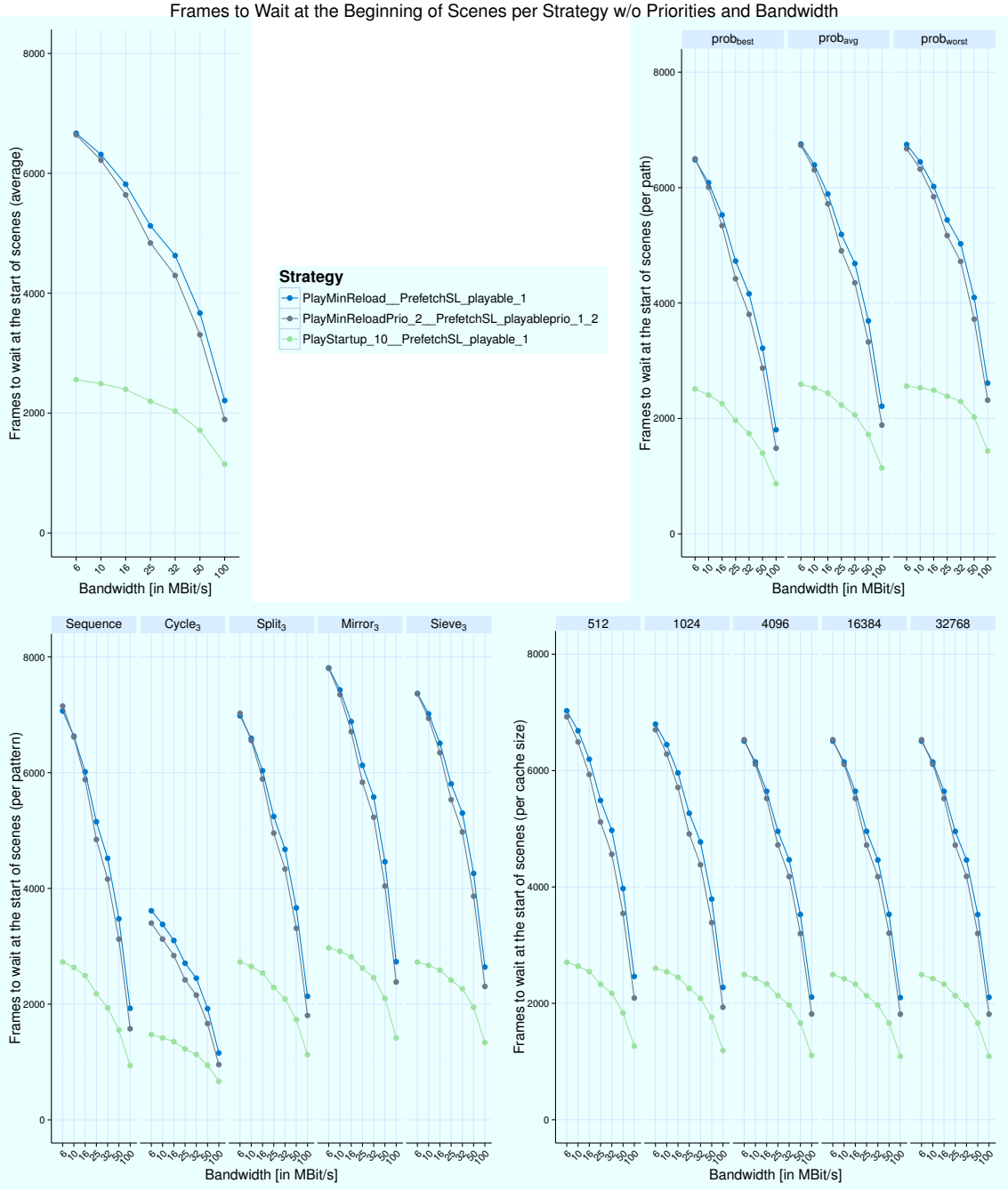


Figure I.44.: Evaluation of the priority-based strategies (detailed results) - frames to wait before playback for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

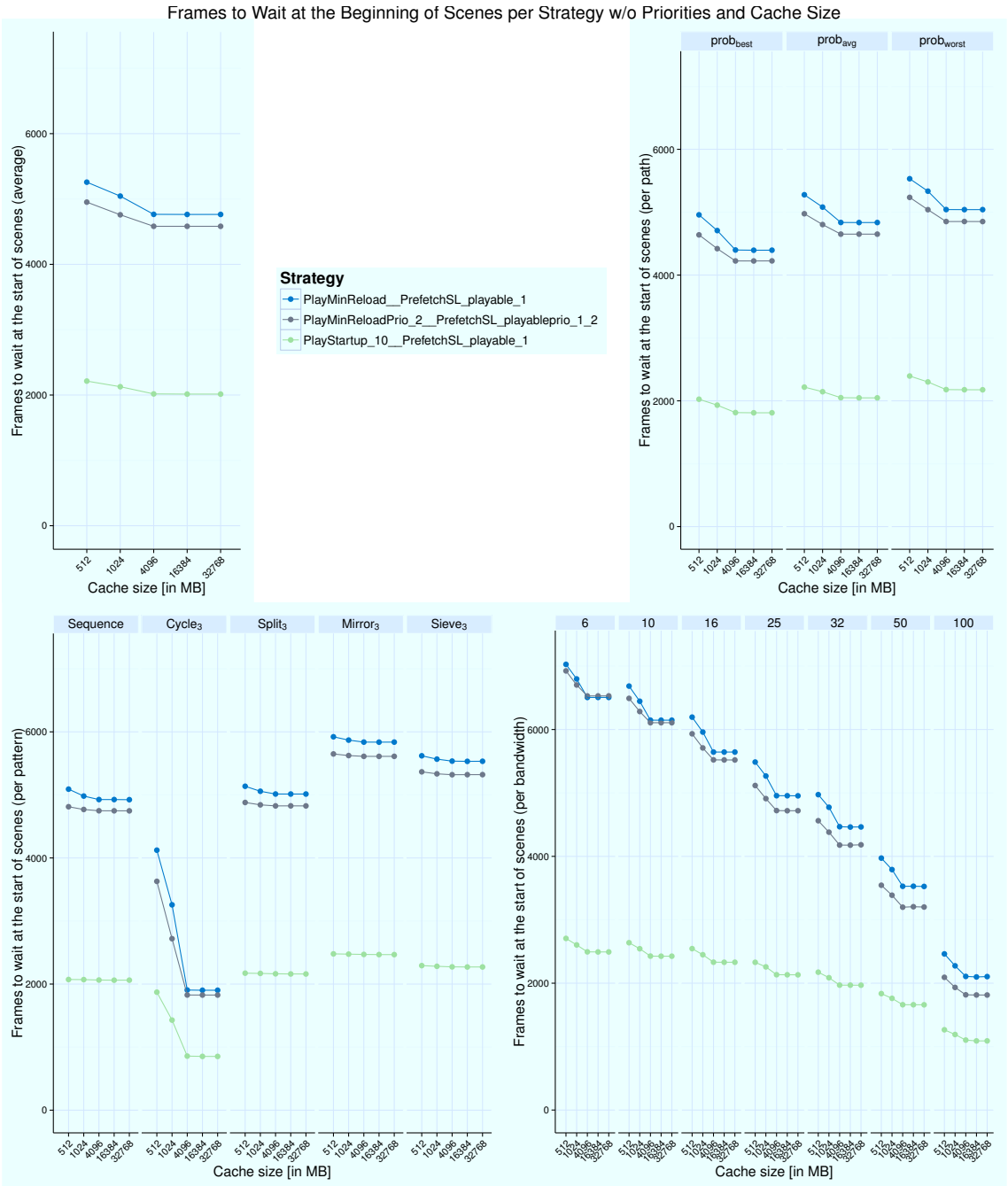


Figure I.45.: Evaluation of the priority-based strategies (detailed results) - frames to wait before playback for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

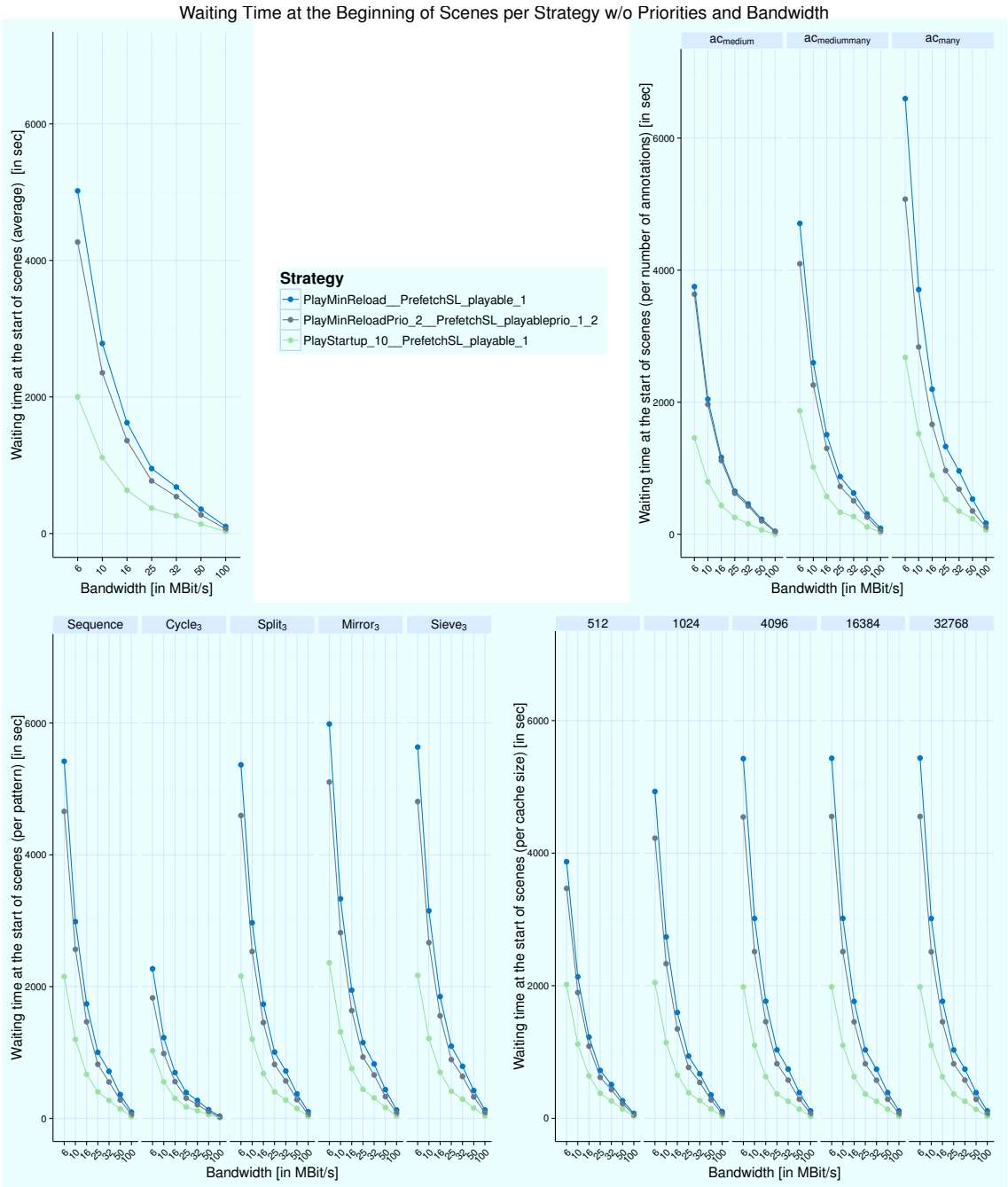


Figure I.46.: Evaluation of the priority-based strategies (detailed results) - waiting time before playback for different bandwidths: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

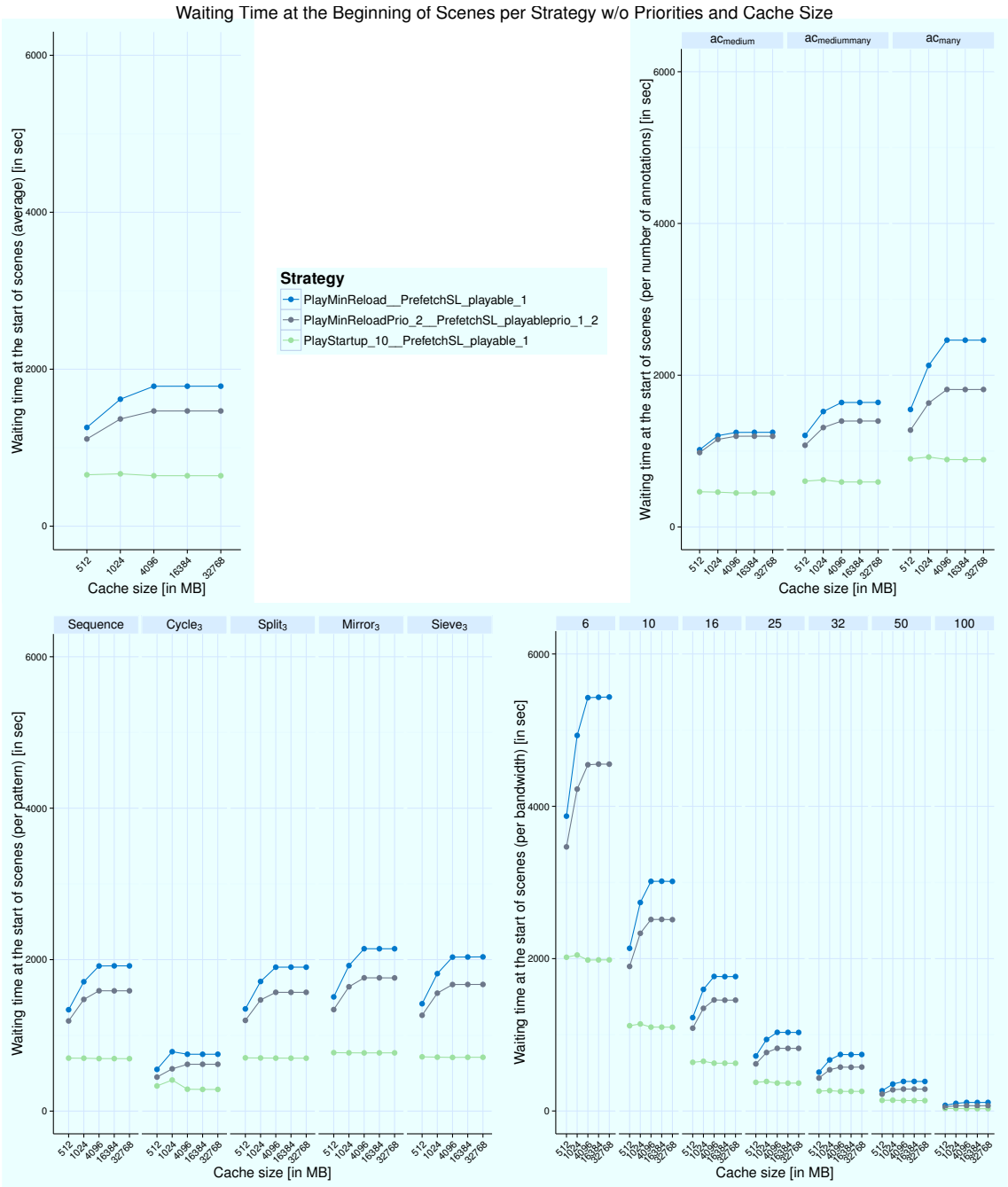


Figure I.47.: Evaluation of the priority-based strategies (detailed results) - waiting time before playback for different cache sizes: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

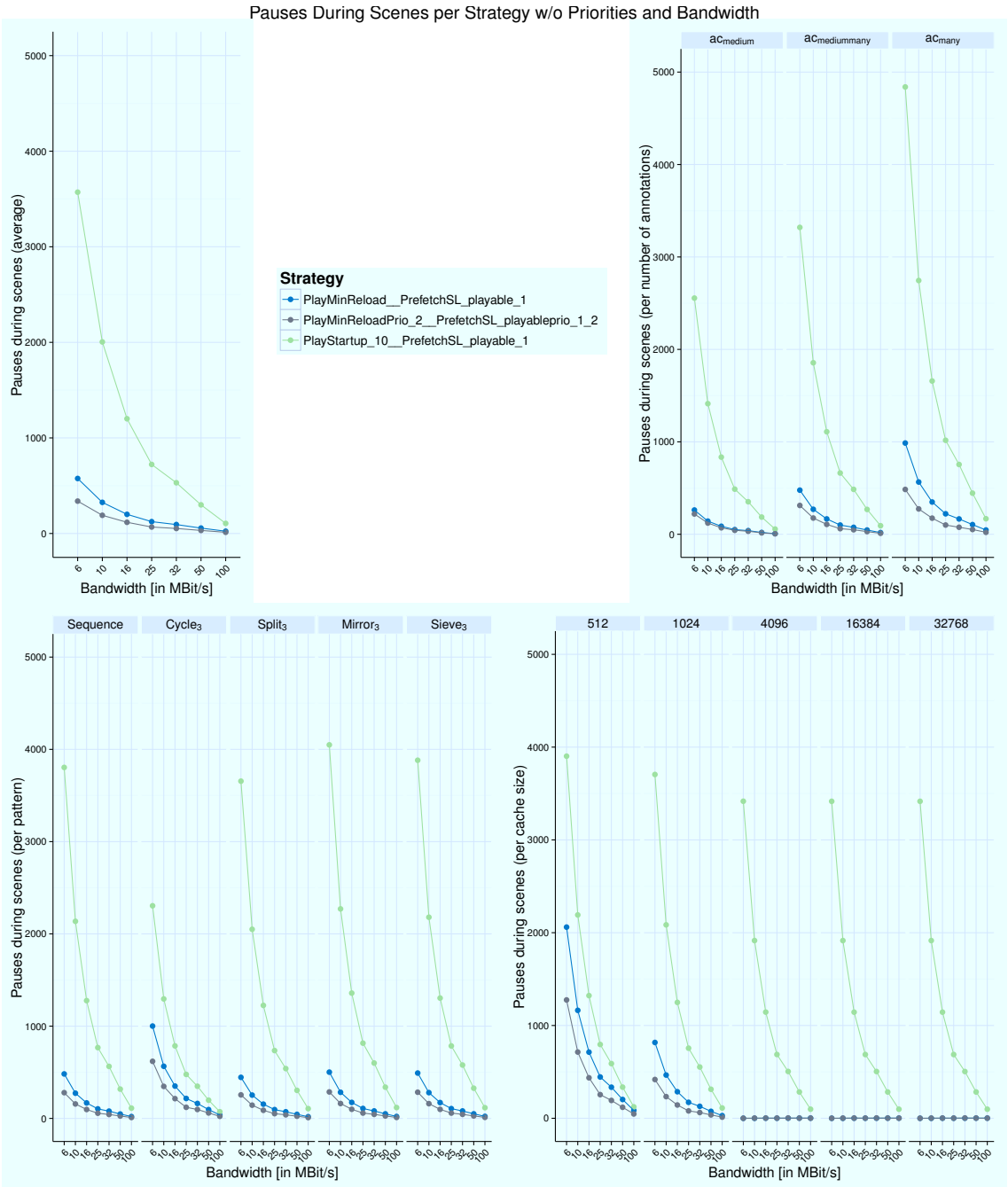


Figure I.48.: Evaluation of the priority-based strategies (detailed results) - pauses during playback for different bandwidths: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

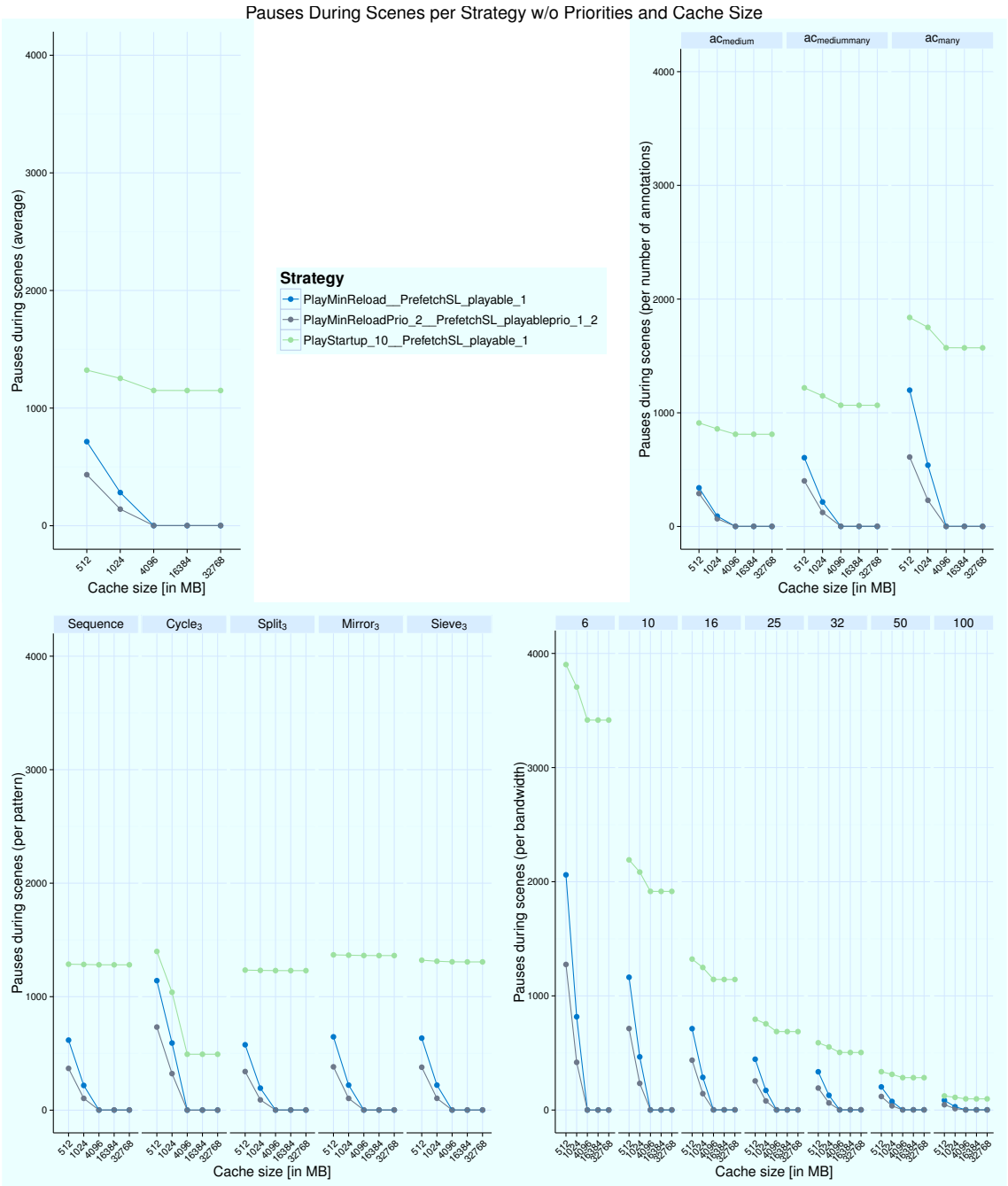


Figure I.49.: Evaluation of the priority-based strategies (detailed results) - pauses during playback for different cache sizes: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

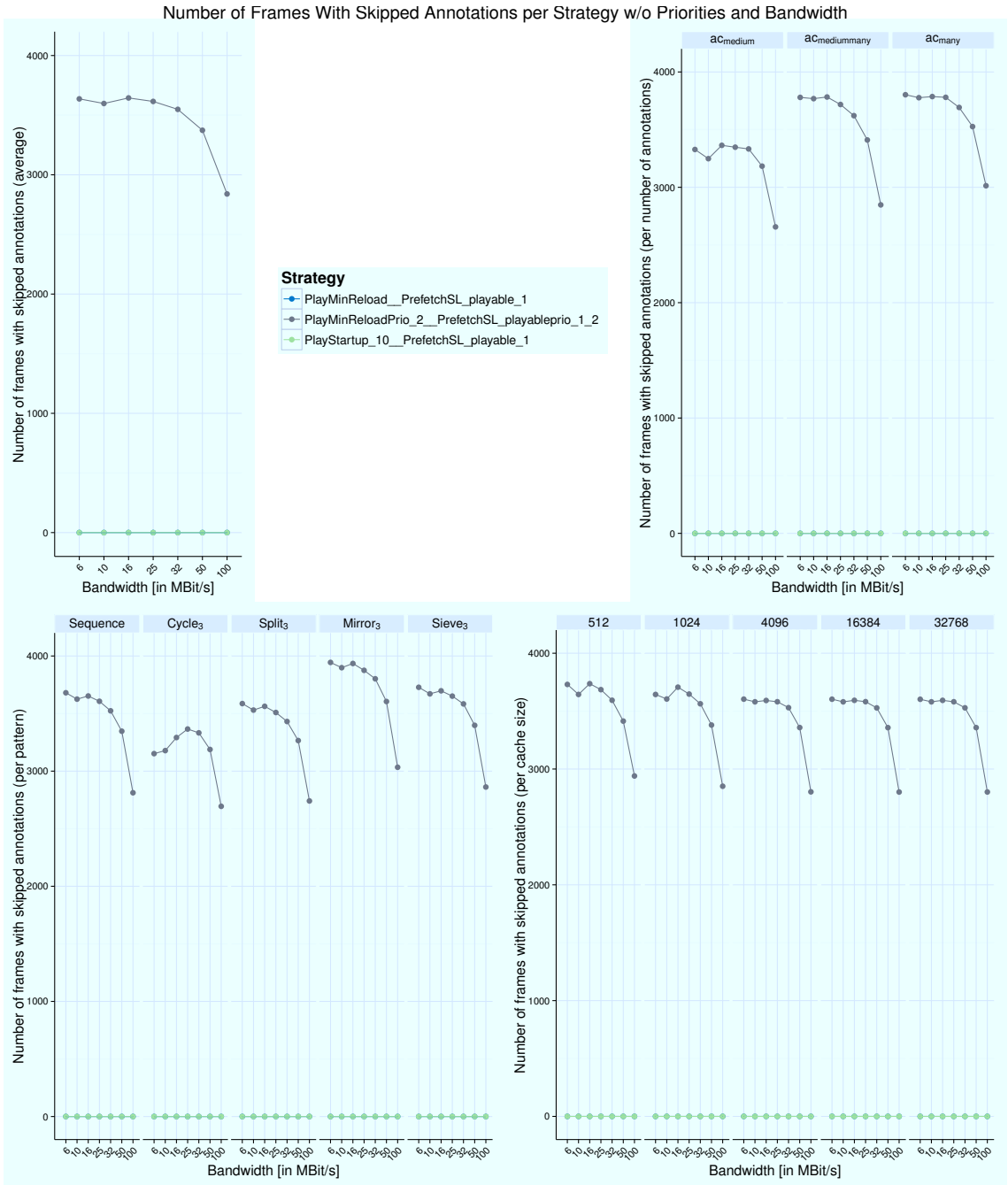


Figure I.50.: Evaluation of the priority-based strategies (detailed results) - number of skipped elements for different bandwidths: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

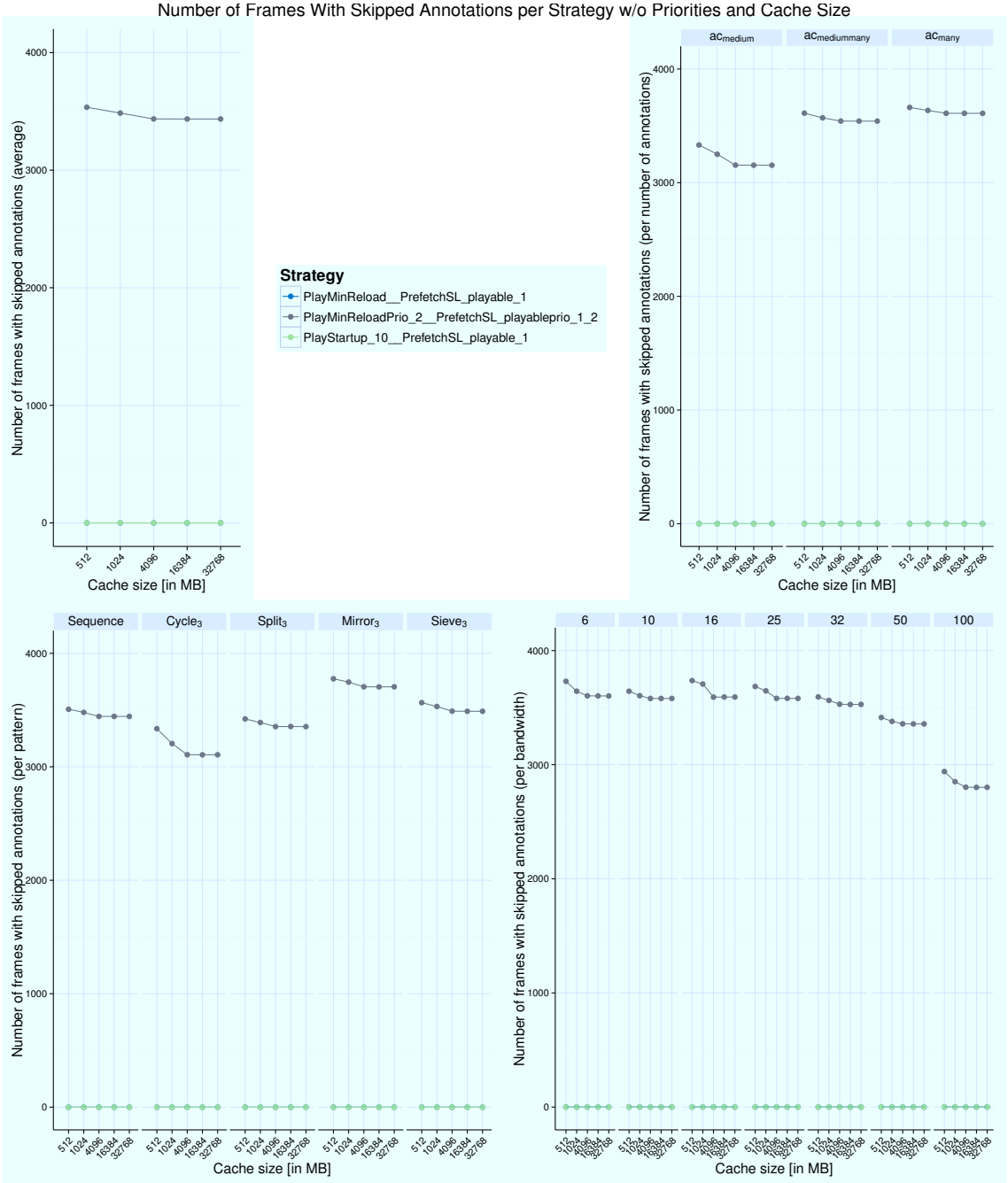


Figure I.51.: Evaluation of the priority-based strategies (detailed results) - number of skipped elements for different cache sizes: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

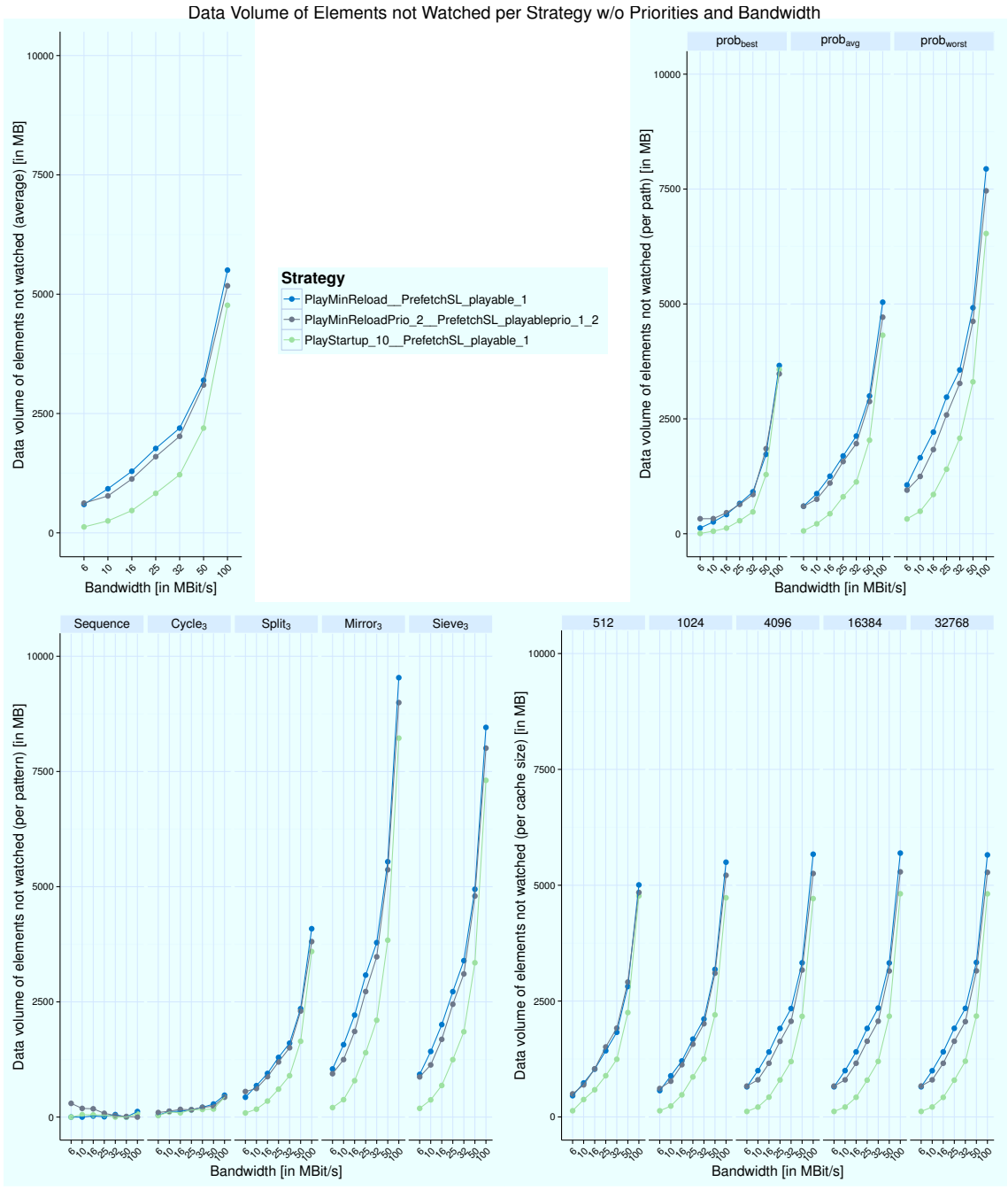


Figure I.52.: Evaluation of the priority-based strategies (detailed results) - data volume of elements not watched for different bandwidths: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

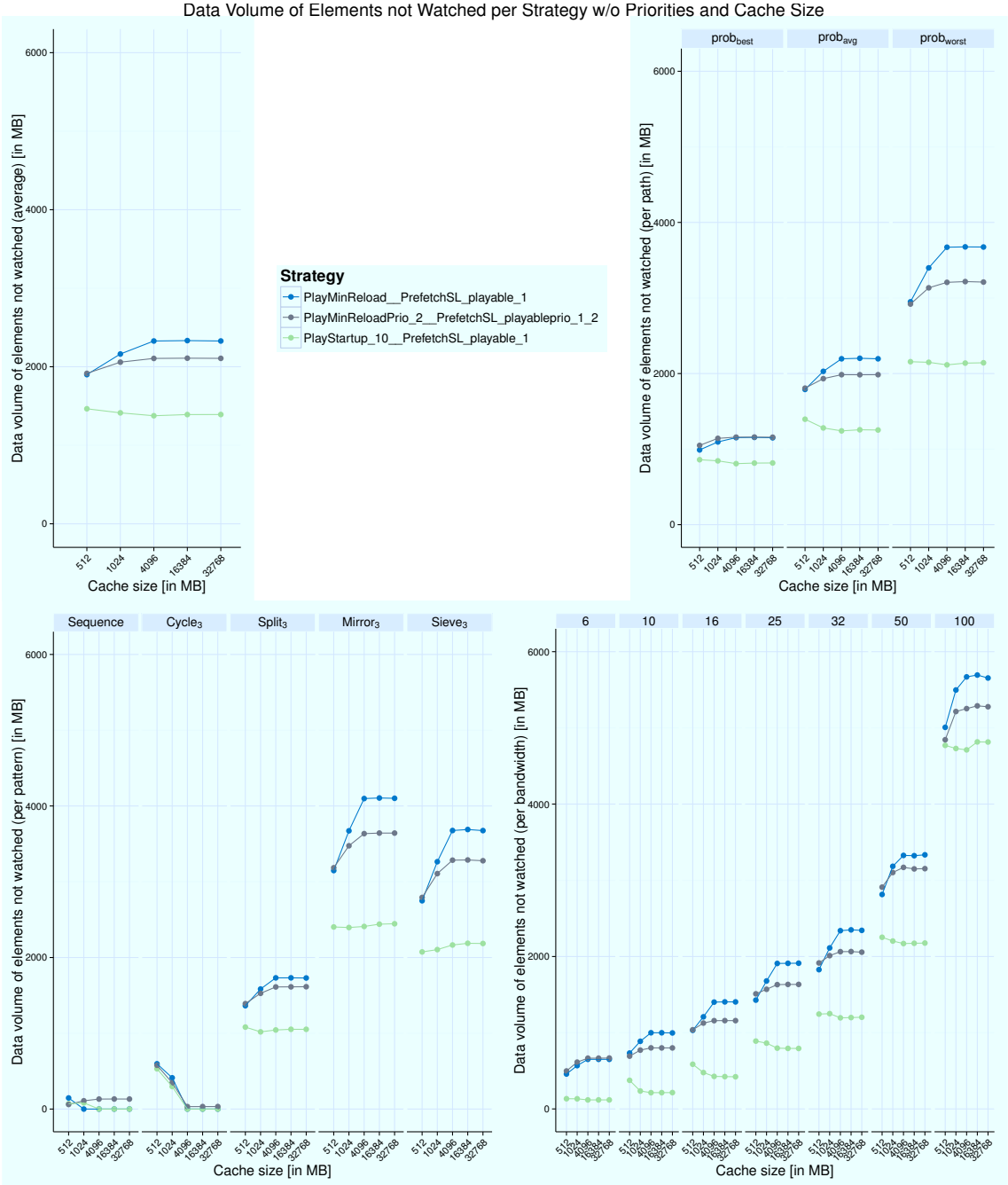


Figure I.53.: Evaluation of the priority-based strategies (detailed results) - data volume of elements not watched for different cache sizes: average for the whole test (upper left), results grouped by used probabilities (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

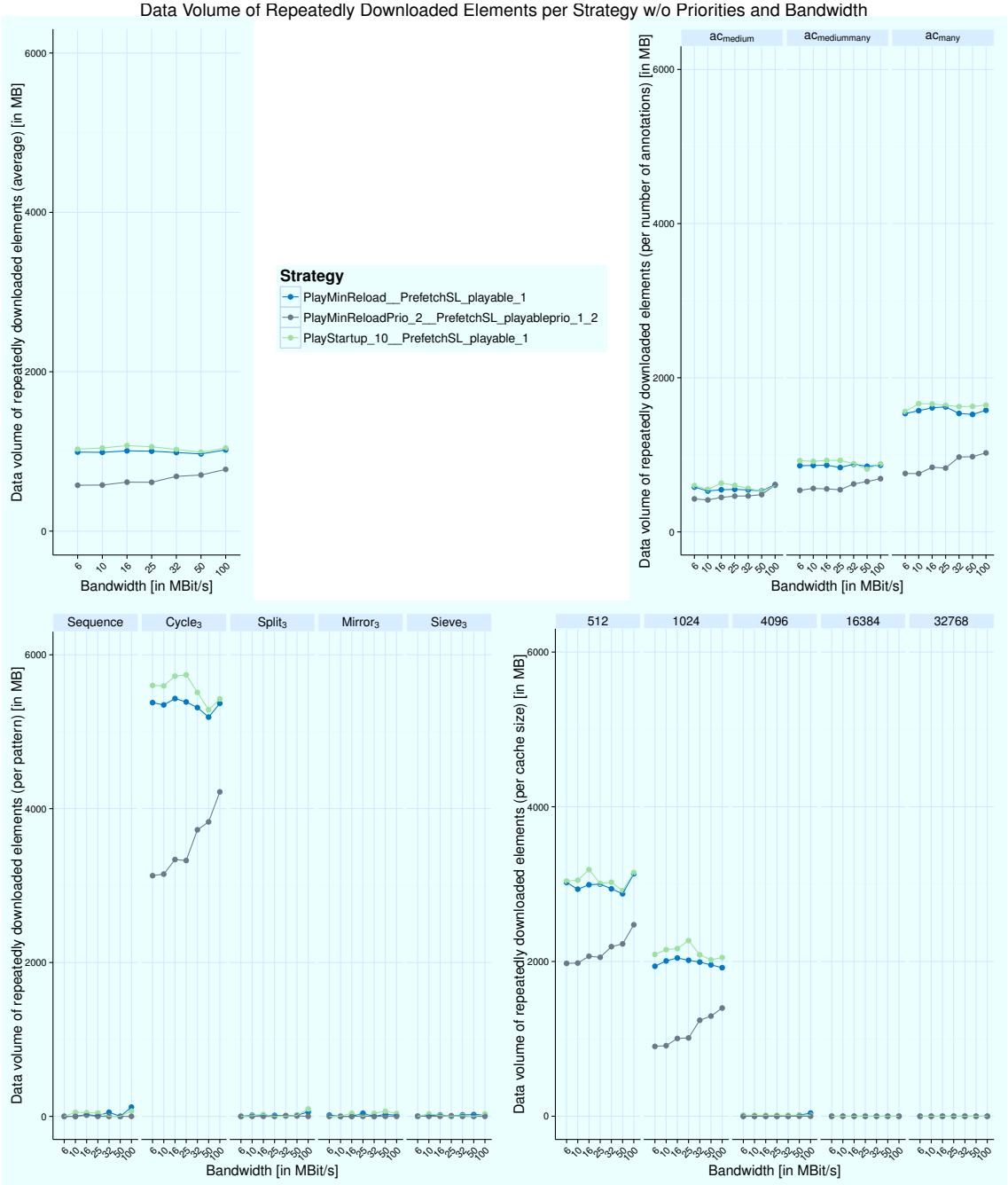


Figure I.54.: Evaluation of the priority-based strategies (detailed results) - data volume of repeated downloads for different bandwidths: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

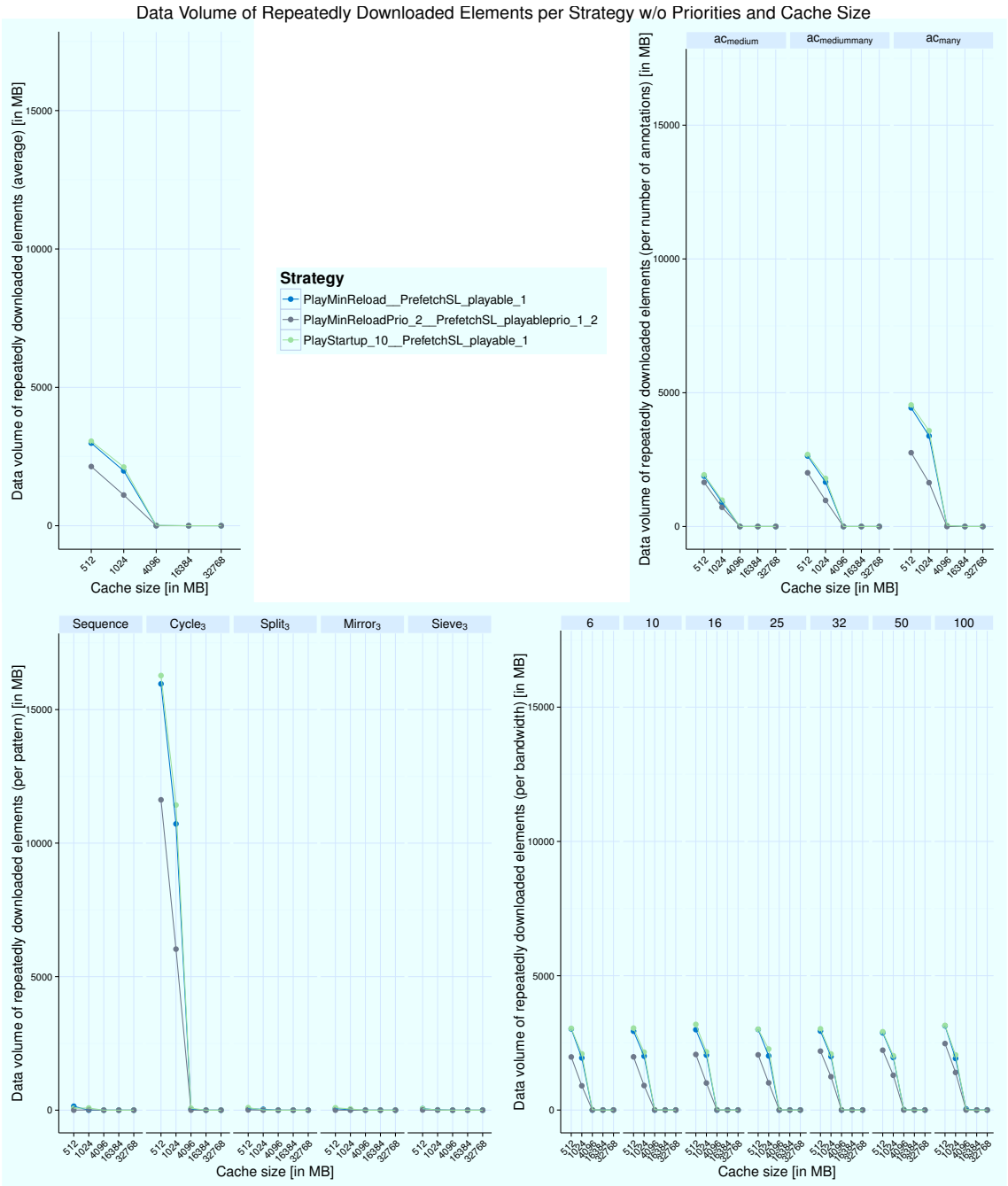


Figure I.55.: Evaluation of the priority-based strategies (detailed results) - data volume of repeated downloads for different cache sizes: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

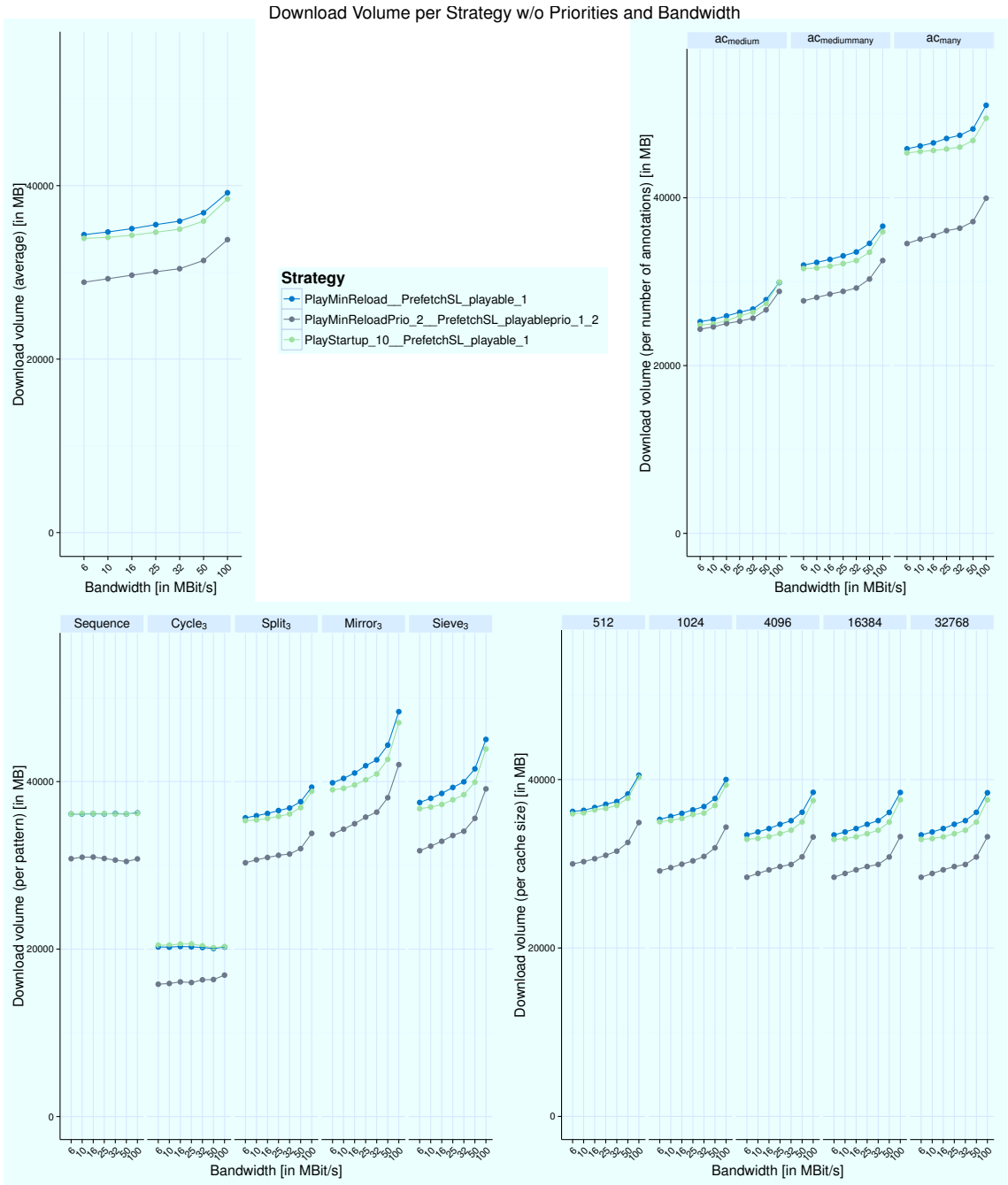


Figure I.56.: Evaluation of the priority-based strategies (detailed results) - download volume for different bandwidths: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by cache size (lower right).

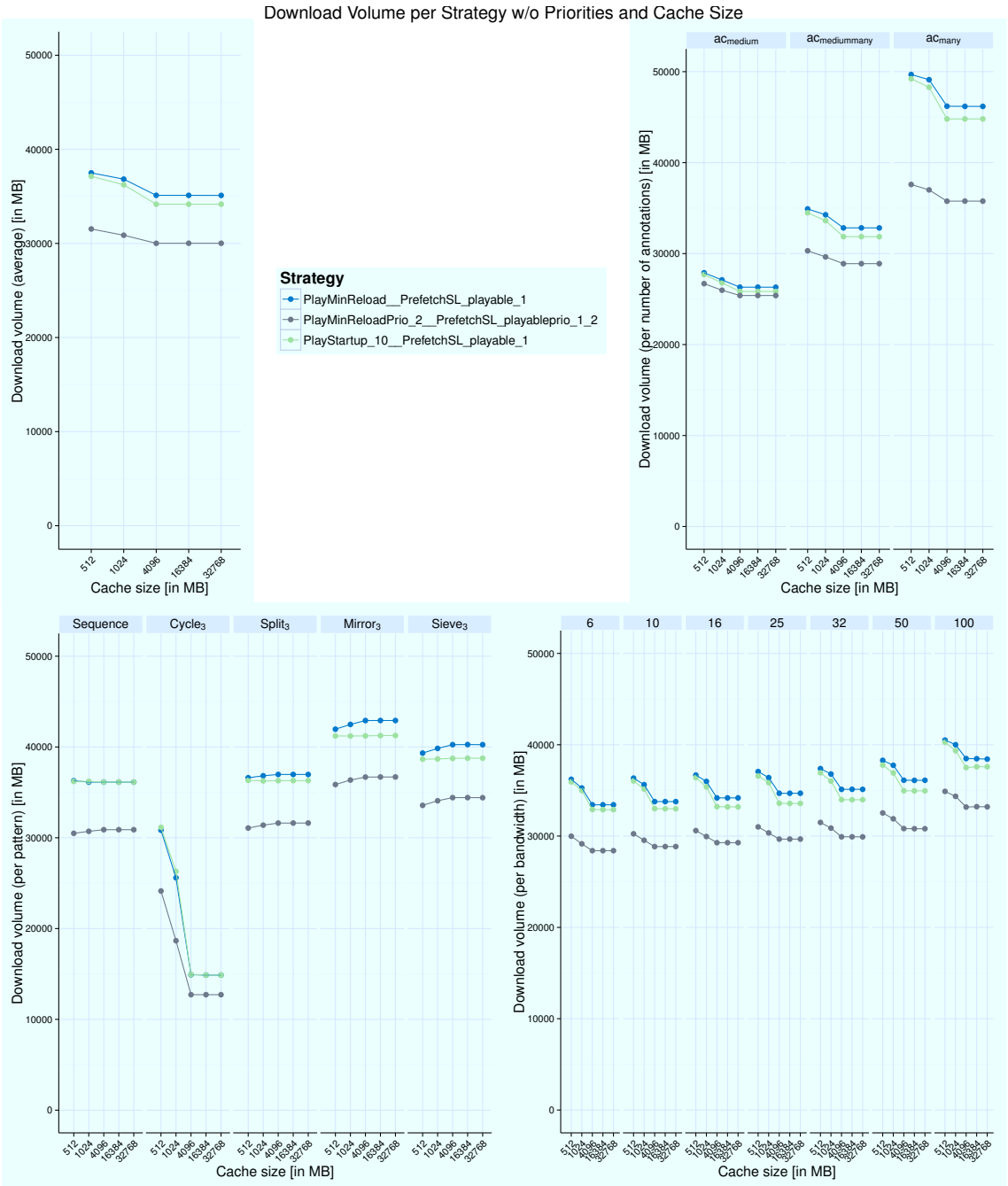


Figure I.57.: Evaluation of the priority-based strategies (detailed results) - download volume for different cache sizes: average for the whole test (upper left), results grouped by the number of annotations (upper right), results grouped by pattern (lower left), and results grouped by bandwidth (lower right).

Bibliography

[5mi14] 5min Media Ltd. *5minMedia VIDEO EVERYWHERE*. Website (accessed January 1, 2014). 2014. URL: http://www.5minmedia.com/.

[Abd08] A. Abdelli. "Extending the verification of multimedia presentation consistency to resource requirements". In: *2nd International Conference on Future Generation Communication and Networking, 2008. FGCN '08.* Vol. 1. IEEE, 2008, pp. 216–219. DOI: 10.1109/FGCN.2008.125.

[ADM06] A. Abhari, S. P. Dandamudi, and S. Majumdar. "Web object-based storage management in proxy caches". In: *Future Generation Computer Systems* 22.1-2 (2006), pp. 16–31. DOI: 10.1016/j.future.2005.08.003.

[Abr+95] M. Abrams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox. *Caching proxies: limitations and potentials*. Technical Report. Website (accessed January 1, 2014). Blacksburg, VA, USA: Virginia Polytechnic Institute & State University, 1995. URL: http://www.w3.org/Conferences/WWW4/Papers/155/.

[Abr+96] M. Abrams, C. R. Standridge, G. Abdulla, E. A. Fox, and S. Williams. "Removal policies in network caches for World-Wide Web documents". In: *ACM SIGCOMM Computer Communication Review* 26.4 (1996), pp. 293–305. DOI: 10.1145/248157.248182.

[Aca07] Academic Technology Services. University of Minnesota. *VideoANT*. Website (accessed April 26, 2014). 2007. URL: http://ant.umn.edu/.

[ASS99] S. Adali, M. L. Sapino, and V. S. Subrahmanian. "A multimedia presentation algebra". In: *SIGMOD Rec.* 28.2 (1999), pp. 121–132. DOI: 10.1145/304181.304193.

[ASS00] S. Adali, M. L. Sapino, and V. S. Subrahmanian. "An algebra for creating and querying multimedia presentations". In: *Multimedia Systems* 8.3 (2000), pp. 212–230. DOI: 10.1007/s005300000046.

[Ada+00] S. Adali, L. Console, M. L. Sapino, M. Schenone, and P. Terenziani. "Representing and reasoning with temporal constraints in multimedia presentations". In: *Proceedings. Seventh International Workshop on Temporal Representation and Reasoning, 2000. TIME 2000.* 2000, pp. 3–11. DOI: 10.1109/TIME.2000.856576.

[AZM99] M. Adiba and J.-L. Zechinelli-Martini. "Spatio-temporal multimedia presentations as database objects". In: *Database and Expert Systems Applications*. Ed. by T. J. Bench-Capon, G. Soda, and A. M. Tjoa. Vol. 1677. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, pp. 974–985. DOI: 10.1007/3-540-48309-8_92.

[AL95] D. A. Adjeroh and M. C. Lee. “Synchronization mechanisms for distributed multimedia presentation systems”. In: *Proceedings. International Workshop on Multi-Media Database Management Systems, 1995*. IEEE, 1995, pp. 30–37. DOI: 10.1109/MMDBMS.1995.520420.

[Adv09] Advanced Distributed Learning (ADL). *Sharable Content Object Reference Model (SCORM), 2004, 4th Edition, Testing Requirements Version 1.1*. Standard/Website (accessed April 26, 2014). 2009. URL: http://www.adlnet.gov/wp-content/uploads/2011/07/SCORM_2004_4ED_v1_1_TR_20090814.pdf.

[AWY99] C. Aggarwal, J. L. Wolf, and P. S. Yu. “Caching on the World Wide Web”. In: *IEEE Transactions on Knowledge and Data Engineering* 11.1 (1999), pp. 94–107. DOI: 10.1109/69.755618.

[AT03] K. Al-Tahat. “The development of a framework for multimedia courseware”. In: *Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003*. Vol. 1. 2003, pp. 236–239. DOI: 10.1109/ICECS.2003.1302020.

[All+08] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov. “A survey of scheduling problems with setup times or costs”. In: *European Journal of Operational Research* 187.3 (2008), pp. 985–1032. DOI: 10.1016/j.ejor.2006.06.060.

[All83] J. F. Allen. “Maintaining knowledge about temporal intervals”. In: *Communications of the ACM* 26.11 (1983), pp. 832–843. DOI: 10.1145/182.358434.

[ABM07] H. Alustwani, J. M. Bahi, and A. Mostefaoui. “Managing VCR interactions in multimedia presentations”. In: *1st International Conference on Digital Information Management, 2006*. 2007, pp. 179–186. DOI: 10.1109/ICDIM.2007.369350.

[And] Android Developers. *Performance tips*. Website (accessed April 26, 2014). URL: http://developer.android.com/training/articles/perf-tips.html.

[AW97] M. F. Arlitt and C. L. Williamson. “Internet Web servers: workload characterization and performance implications”. In: *IEEE/ACM Transactions on Networking* 5.5 (1997), pp. 631–645. DOI: 10.1109/90.649565.

[Arl+00] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich, and T. Jin. “Evaluating content management techniques for Web proxy caches”. In: *ACM SIGMETRICS Performance Evaluation Review* 27.4 (2000), pp. 3–11. DOI: 10.1145/346000.346003.

[AD05] M. K. Asadi and J.-C. Dufourd. “Context-aware semantic adaptation of multimedia presentations”. In: *IEEE International Conference on Multimedia and Expo, 2005. ICME 2005*. IEEE, 2005, pp. 362–365. DOI: 10.1109/ICME.2005.1521435.

[Ass99] N. A. Assimakopoulos. “A generic spatial temporal computation model for systemic multimedia presentations”. In: *Cybernetics and Systems* 30.6 (1999), pp. 509–531. DOI: 10.1080/019697299125082.

[AP05] O. Aubert and Y. Prié. “Advene: active reading through hypervideo”. In: *Proceedings of the 16th ACM Conference on Hypertext and Hypermedia. HYPERTEXT ’05*. New York, NY, USA: ACM, 2005, pp. 235–244. DOI: 10.1145/1083356.1083405.

[AP07] O. Aubert and Y. Prié. “Advene: an open-source framework for integrating and visualising audiovisual metadata”. In: *Proceedings of the 15th International Conference on Multimedia. MULTIMEDIA ’07*. New York, NY, USA: ACM, 2007, pp. 1005–1008. DOI: 10.1145/1291233.1291451.

[APS12] O. Aubert, Y. Prié, and D. Schmitt. “Advene as a tailorable hypervideo authoring tool: a case study”. In: *Proceedings of the 2012 ACM Symposium on Document Engineering. DocEng ’12*. New York, NY, USA: ACM, 2012, pp. 79–82. DOI: 10.1145/2361354.2361370.

[Aub+08] O. Aubert, P.-A. Champin, Y. Prié, and B. Richard. “Canonical processes in active reading and hypervideo production”. In: *Multimedia Systems* 14.6 (2008), pp. 427–433. DOI: 10.1007/s00530-008-0132-2.

[Avr+11a] Z. Avramova, D. Vleeschauwer, S. Wittevrongel, and H. Bruneel. “Performance analysis of a caching algorithm for a catch-up television service”. In: *Multimedia Systems* 17.1 (2011), pp. 5–18. DOI: 10.1007/s00530-010-0201-1.

[Avr+11b] Z. Avramova, D. Vleeschauwer, P. Debevere, S. Wittevrongel, P. Lambert, R. Walle, and H. Bruneel. “On the performance of scalable video coding for VBR TV channels transport in multiple resolutions and qualities”. In: *Multimedia Tools and Applications* 53.3 (2011), pp. 487–517. DOI: 10.1007/s11042-010-0506-2.

[AZ01a] R. S. Aygün and A. Zhang. “Interactive multimedia presentation management in distributed multimedia systems”. In: *Proceedings International Conference on Information Technology: Coding and Computing, 2001*. IEEE, 2001, pp. 275–279. DOI: 10.1109/ITCC.2001.918805.

[AZ01b] R. S. Aygün and A. Zhang. “Middle-tier for multimedia synchronization”. In: *Proceedings of the 9th ACM International Conference on Multimedia. MULTIMEDIA ’01*. New York, NY, USA: ACM, 2001, pp. 471–474. DOI: 10.1145/500141.500215.

[AZ02] R. S. Aygün and A. Zhang. “Modeling and verification of interactive flexible multimedia presentations using PROMELA/SPIN”. In: *Model Checking Software*. Ed. by D. Bosnacki and S. Leue. Vol. 2318. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 205–212. DOI: 10.1007/3-540-46017-9_15.

[Bah+02] H. Bahn, K. Koh, S. H. Noh, and S. M. Lyul. “Efficient replacement of nonuniform objects in Web caches”. In: *Computer* 35.6 (2002), pp. 65–73. DOI: 10.1109/MC.2002.1009170.

[Bai+98] B. Bailey, J. A. Konstan, R. Cooley, and M. Dejong. “Nsync - a toolkit for building interactive multimedia presentations”. In: *Proceedings of the 6th ACM International Conference on Multimedia. MULTIMEDIA ’98*. New York, NY, USA: ACM, 1998, pp. 257–266. DOI: 10.1145/290747.290779.

[Bal+01] D. Balfanz, M. Finke, C. Jung, and R. Wichert. “An interactive video system supporting e-commerce product placement”. In: *World Scientific and Engineering Society -WSES-: Proceedings of WSES Conferences 2001*. Website (accessed April 26, 2014). 2001, Paper 255. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.3305&rep=rep1&type=pdf.

[BO98] N. H. Balkir and G. Ozsoyoglu. “Multimedia presentation servers: buffer management and admission control”. In: *Proceedings International Workshop on Multi-Media Database Management Systems, 1998*. IEEE, 1998, pp. 154–161. DOI: 10.1109/MMDBMS.1998.709778.

[BBO13] L. B. Baltussen, J. Blom, and R. Ordelman. “VideoHypE: an editor tool for supervised automatic video hyperlinking”. In: *Intelligent Technologies for Interactive Entertainment*. Ed. by M. Mancas, N. d’Alessandro, X. Siebert, B. Gosselin, C. Valderrama, and T. Dutoit. Vol. 124. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer International Publishing, 2013, pp. 43–48. DOI: 10.1007/978-3-319-03892-6_5.

[BSR10] R. Bartak, M. A. Salido, and F. Rossi. “New trends in constraint satisfaction, planning, and scheduling: a survey”. In: *Knowledge Engineering Review* 25.3 (2010), pp. 249–279. DOI: 10.1017/S0269888910000202.

[Bea+02] R. Beales, D. Cruickshank, D. DeRoure, N. Gibbins, B. Juby, D. T. Michaelides, and K. R. Page. “The pipeline of enrichment: supporting link creation for continuous media”. In: *Hypermedia: Openness, Structural Awareness, and Adaptivity*. Ed. by S. Reich, M. M. Tzagarakis, and P. M. Bra. Vol. 2266. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 47–58. DOI: 10.1007/3-540-45844-1_5.

[BP10] A. Becker and M. Pant. *Android 2 - Grundlagen und Programmierung*. 2. Heidelberg: dpunkt.verlag, 2010.

[Bee10] Beet Media LLC. *“Clickable” videos finding traction with big brand marketers*. Website (accessed April 26, 2014). 2010. URL: http://www.beet.tv/2010/03/clickablevideo.html.

[Bel12] Belldon Limited. *Quicktvpro*. Website (accessed April 26, 2014). 2012. URL: http://quicktvpro.com/.

[BZ96a] J. C. R. Bennett and H. Zhang. “Hierarchical packet fair queueing algorithms”. In: *ACM SIGCOMM Computer Communication Review* 26.4 (1996), pp. 143–156. DOI: 10.1145/248157.248170.

[BZ96b] J. C. R. Bennett and H. Zhang. “WF2Q: worst-case fair weighted fair queueing”. In: *Proceedings IEEE INFOCOM ’96. 15th Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation*. Vol. 1. IEEE, 1996, pp. 120–128. DOI: 10.1109/INFCOM.1996.497885.

[Ber12a] E. Berndl. “SIVA-XML-Schema vs. SMIL - Strukturmapping und Exportimplementierung”. Master’s thesis. Passau, Germany: Universität Passau, 2012.

[Ber98] M. Bernstein. “Patterns of hypertext”. In: *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space - Structure in Hypermedia Systems - HYPERTEXT ’98*. New York, NY, USA: ACM, 1998, pp. 21–29. DOI: 10.1145/276627.276630.

[Ber+05] E. Bertino, E. Ferrari, A. Perego, and D. Santi. “A constraint-based approach for the authoring of multi-topic multimedia presentations”. In: *IEEE International Conference on Multimedia and Expo, 2005. ICME 2005.* IEEE, 2005, pp. 578–581. DOI: 10.1109/ICME.2005.1521489.

[BFS00] E. Bertino, E. Ferrari, and M. Stolf. “MPGS: an interactive tool for the specification and generation of multimedia presentations”. In: *IEEE Transactions on Knowledge and Data Engineering* 12.1 (2000), pp. 102–125. DOI: 10.1109/69.842254.

[Ber12b] P. Bertolino. “Sensarea: an authoring tool to create accurate clickable videos”. In: *10th International Workshop on Content-Based Multimedia Indexing (CBMI), 2012*. IEEE, 2012, pp. 1–4. DOI: 10.1109/CBMI.2012.6269804.

[BR02] F. Bes and C. Roisin. “A presentation language for controlling the formatting process in multimedia presentations”. In: *Proceedings of the 2002 ACM Symposium on Document Engineering. DocEng ’02*. New York, NY, USA: ACM, 2002, pp. 2–9. DOI: 10.1145/585058.585061.

[BK04] N. Bevan and S. Kincla. *HCI design foundation study: final report*. Website (accessed April 26, 2014). London, 2004. URL: http://www.jisc.ac.uk/whatwedo/programmes/presentation/hcidesign.aspx.

[Bhi+10] W. Bhikharie, H. Huurdeman, M. van de Watering, and E. Anton. *XIMPEL overview*. Website (accessed April 26, 2014). 2010. URL: http://www.ximpel.org/.

[BC08] N. Bian and H. Chen. “A least grade page replacement algorithm for web cache optimization”. In: *1st International Workshop on Knowledge Discovery and Data Mining, 2008. WKDD ’08.* IEEE, 2008, pp. 469–472. DOI: 10.1109/WKDD.2008.124.

[Bia+13] S. Bianco, G. Ciocca, P. Napoletano, R. Schettini, R. Margherita, G. Marini, and G. Pantaleo. “Cooking action recognition with iVAT: an interactive video annotation tool”. In: *Image Analysis and Processing - ICIAP 2013*. Ed. by A. Petrosino. Vol. 8157. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 631–641. DOI: 10.1007/978-3-642-41184-7_64.

[BGVO05a] I. M. Bilasco, J. Gensel, and M. Villanova-Oliver. “STAMP: adaptable templates for synchronized multimedia presentations”. In: *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence, 2005. Proceedings.* IEEE, 2005, pp. 645–648. DOI: 10.1109/WI.2005.137.

[BGVO05b] I. M. Bilasco, J. Gensel, and M. Villanova-Oliver. “STAMP: a model for generating adaptable multimedia presentations”. In: *Multimedia Tools and Applications* 25.3 (2005), pp. 361–375. DOI: 10.1007/s11042-005-6540-9.

[BS96] G. Blakowski and R. Steinmetz. “A media synchronization survey: reference model, specification, and case studies”. In: *IEEE Journal on Selected Areas in Communications* 14.1 (1996), pp. 5–35. DOI: 10.1109/49.481691.

[BHL92] G. Blakowski, J. Hübel, and U. Langrehr. “Tools for specifying and executing synchronized multimedia presentations”. In: *Network and Operating System Support for Digital Audio and Video*. Ed. by R. Herrtwich. Vol. 614. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1992, pp. 271–282. DOI: 10.1007/3-540-55639-7_24.

[BF05] M. A. Bochicchio and N. Fiore. “Teacher-centered production of hypervideo for distance learning”. In: *International Journal of Distance Education Technologies (IJDET)* 3.4 (2005), pp. 19–34. DOI: 10.4018/jdet.2005100103.

[BK01] S. Boll and W. Klas. “ZYX - a multimedia document model for reuse and adaptation of multimedia content”. In: *IEEE Transactions on Knowledge and Data Engineering* 13.3 (2001), pp. 361–382. DOI: 10.1109/69.929895.

[BKW99] S. Boll, W. Klas, and U. Westermann. “Exploiting ORDBMS technology to implement the ZYX data model for multimedia documents and presentations”. In: *Datenbanksysteme in Büro, Technik und Wissenschaft*. Ed. by A. P. Buchmann. Informatik aktuell. Springer Berlin Heidelberg, 1999, pp. 232–250. DOI: 10.1007/978-3-642-60119-4_14.

[BKW00] S. Boll, W. Klas, and U. Westermann. “Multimedia document models: sealed fate or setting out for new shores?” In: *Multimedia Tools and Applications* 11.3 (2000), pp. 267–279. DOI: 10.1023/A:1009606112260.

[BH96] J.-C. Bolot and P. Hoschka. “Performance engineering of the World Wide Web: application to dimensioning and cache design”. In: *Computer Networks and ISDN Systems* 28.7-11 (1996), pp. 1397–1405. DOI: 10.1016/0169-7552(96)00073-6.

[BDV09] E. Bomcke and C. De Vleeschouwer. “An interactive video streaming architecture for H.264/AVC compliant players”. In: *IEEE International Conference on Multimedia and Expo, 2009. ICME 2009*. IEEE, 2009, pp. 1554–1555. DOI: 10.1109/ICME.2009.5202804.

[Bor+96] M. Bordegoni, G. Faconti, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. “Intelligent multimedia presentation systems: a proposal for a reference model”. In: *Multimedia Modeling: Towards the Information Superhighway*. Website (accessed April 26, 2014). World Scientific, 1996, pp. 3–20. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.6769.

[Bor+97] M. Bordegoni, G. Faconti, S. Feiner, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. “A standard reference model for intelligent multimedia presentation system”. In: *Computer Standards & Interfaces* 18.6-7 (1997), pp. 477–496. DOI: 10.1016/S0920-5489(97)00013-5.

[BMV10] F. Boronat, M. Montagud, and V. Vidal. “Enhanced RTP-based tool-set for video streaming simulation using NS-2”. In: *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia. MoMM ’10*. Vol. 2. New York, NY, USA: ACM, 2010, pp. 382–386. DOI: 10.1145/1971519.1971584.

[Bos89] J. J. Bosco. “Interactive Video: Educational Tool or Toy?” In: *Interactive Video. Volume 1 of The Educational Technology Anthology Series*. Ed. by C. Educational Technology Publications. illustrate. Englewood Cliffs, NJ, USA: Educational Technology Publications, 1989. Chap. 1, pp. 3–9.

[BCF02] F. Bota, F. Corno, and L. Farinetti. “Hypervideo: a parameterized hotspot approach”. In: *ICWI2002: IIADIS International Conference WWW/Internet 2002*. Ed. by P. Isaias. IADIS - International Association for Development of the Information Society, 2002, pp. 620–623.

[BB11] S. Bouyakoub and A. Belkhir. “SMIL Builder: an incremental authoring tool for SMIL documents”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 7.1 (2011), 2:1–2:30. DOI: 10.1145/1870121.1870123.

[BET99] P. Branch, G. Egan, and B. Tonkin. “A client caching scheme for interactive video-on-demand”. In: *Proceedings. IEEE International Conference on Networks, 1999. (ICON ’99)*. IEEE, 1999, pp. 391–397. DOI: 10.1109/ICON.1999.796202.

[BD89] H. P. Brondmo and G. Davenport. “Creating and viewing the Elastic Charles - a hypermedia journal”. In: *Hypertext: State of the ART, Papers Presented at the Hypertext II Conference*. Ed. by R. McAlesse and C. Green. Website (accessed April 26, 2014). York, England: Intellect, 1989, pp. 43–51. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.5291&rep=rep1&type=pdf.

[BZ05] M. C. Buchanan and P. T. Zellweger. “Automatic temporal layout mechanisms revisited”. In: *ACM Transactions on Multimedia Computing, Communications and Applications* 1.1 (2005), pp. 60–88. DOI: 10.1145/1047936.1047942.

[BR08] D. C. A. Bulterman and L. W. Rutledge. *SMIL 3.0: flexible multimedia for Web, mobile devices and daisy talking books*. 2nd, illustrated. Radon series on computational and applied mathematics. Springer, 2008.

[Bul93] D. C. A. Bulterman. “Specification and support of adaptable networked multimedia”. In: *Multimedia Systems* 1.2 (1993), pp. 68–76. DOI: 10.1007/BF01213485.

[Bul03] D. C. A. Bulterman. “Using SMIL to encode interactive, peer-level multimedia annotations”. In: *Proceedings of the 2003 ACM Symposium on Document Engineering. DocEng ’03*. New York, NY, USA: ACM, 2003, pp. 32–41. DOI: 10.1145/958227.958228.

[BH05] D. C. A. Bulterman and L. Hardman. “Structured multimedia authoring”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 1.1 (2005), pp. 89–109. DOI: 10.1145/1047936.1047943.

[BRL91] D. C. A. Bulterman, G. V. Rossum, and R. V. Liere. “A structure for transportable, dynamic multimedia documents”. In: *Proceedings of the Summer 1991 USENIX Conference*. Website (accessed April 26, 2014). 1991, pp. 137–156. URL: http://homepages.cwi.nl/~robertl/papers/1991/usenix/paper.pdf.

[Bul+98] D. C. A. Bulterman, L. Hardman, J. Jansen, K. S. Mullender, and L. Rutledge. “GRiNS: A GRaphical INterface for creating and playing SMIL documents”. In: *Computer Networks and ISDN Systems* 30.1-7 (1998), pp. 519–529. DOI: 10.1016/S0169-7552(98)00128-7.

[Bul+04] D. C. A. Bulterman, J. Jansen, K. Kleanthous, K. Blom, and D. Benden. “Ambulant: a fast, multi-platform open source SMIL player”. In: *Proceedings of the 12th Annual ACM International Conference on Multimedia. MULTIMEDIA ’04*. New York, NY, USA: ACM, 2004, pp. 492–495. DOI: 10.1145/1027527.1027646.

[BBY13] G. C. Buttazzo, M. Bertogna, and G. Yao. “Limited preemptive scheduling for real-time systems. A survey”. In: *IEEE Transactions on Industrial Informatics* 9.1 (2013), pp. 3–15. DOI: 10.1109/TII.2012.2188805.

[CWI10] CWI. *AMBULANT open SMIL player (2.4)*. Website (accessed April 26, 2014). 2010. URL: http://www.ambulantplayer.org/.

[CI97] P. Cao and S. Irani. “Cost-aware WWW proxy caching algorithms”. In: *Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems*. Website (accessed April 26, 2014). Monterey, CA, USA: USENIX Association, 1997. URL: https://www.usenix.org/legacy/publications/library/proceedings/usits97/full_papers/cao/cao.pdf.

[Cao+10] Y. Cao, D. Renzel, M. Jarke, R. Klamma, M. Lottko, G. Toubekis, and M. Jansen. “Well-balanced usability & annotation complexity in interactive video semantization”. In: *2010 4th International Conference on Multimedia and Ubiquitous Engineering (MUE)*. IEEE, 2010, pp. 1–8. DOI: 10.1109/MUE.2010.5575051.

[Car01a] D. Carlson. *Modeling XML vocabularies with UML: part I*. Website (accessed April 26, 2014). 2001. URL: http://www.xml.com/pub/a/2001/08/22/uml.html?page=1.

[Car01b] D. Carlson. *Modeling XML vocabularies with UML: part II*. Website (accessed April 26, 2014). 2001. URL: http://www.xml.com/pub/a/2001/09/19/uml.html.

[Car01c] D. Carlson. *Modeling XML vocabularies with UML: part III*. Website (accessed April 26, 2014). 2001. URL: http://www.xml.com/pub/a/2001/10/10/uml.html?page=1.

[Car01d] D. Carlson. *Modeling XML applications with UML: practical e-business applications*. Addison Wesley Publishing Company Incorporated, 2001.

[Car+08] N. Carlsson, A. Mahanti, Z. Li, and D. Eager. “Optimized periodic broadcast of nonlinear media”. In: *IEEE Transactions on Multimedia* 10.5 (2008), pp. 871–884. DOI: 10.1109/TMM.2008.922847.

[Car93] G. S. Carson. “Introduction to the computer graphics reference model”. In: *ACM SIGGRAPH Computer Graphics* 27.2 (1993), pp. 108–119. DOI: 10.1145/165529.165539.

[Cas+91] M. A. Casanova, L. Tucherman, M. J. D. Lima, J. L. Rangel Netto, N. Rodriquez, and L. F. G. Soares. “The nested context model for hyperdocuments”. In: *Proceedings of the 3rd Annual ACM Conference on Hypertext. HYPERTEXT ’91*. December. New York, NY, USA: ACM, 1991, pp. 193–201. DOI: 10.1145/122974.122993.

[Cat+00] R. G. G. Cattell, D. K. Barry, M. Berler, J. Eastman, D. Jordan, C. Russell, O. Schadow, T. Stanienda, and F. Velez, eds. *The Object Data Standard: ODMG 3.0*. The Morgan Kaufmann Series in Data Management Systems Series. Morgan Kaufmann Publishers Inc., 2000.

[CQR11] F. Cazenave, V. Quint, and C. Roisin. “Timesheets.js: when SMIL meets HTML5 and CSS3”. In: *Proceedings of the 11th ACM Symposium on Document Engineering. DocEng ’11*. New York, NY, USA: ACM, 2011, pp. 43–52. DOI: 10.1145/2034691.2034700.

[CG00] A. Celentano and O. Gaggi. “A synchronization model for hypermedia documents navigation”. In: *Proceedings of the 2000 ACM Symposium on Applied Computing - Volume 2. SAC ’00*. New York, NY, USA: ACM, 2000, pp. 585–591. DOI: 10.1145/338407.338508.

[CG03] A. Celentano and O. Gaggi. “Template-based generation of multimedia presentations”. In: *International Journal of Software Engineering and Knowledge Engineering* 13.4 (2003). Website (accessed April 26, 2014), pp. 419–445. URL: http://www.math.unipd.it/~gaggi/doc/jseke.pdf.

[CGS04] A. Celentano, O. Gaggi, and M. L. Sapino. “Retrieval in multimedia presentations”. In: *Multimedia Systems* 10.1 (2004), pp. 72–82. DOI: 10.1007/s00530-004-0138-3.

[CCG01] T. Chambel, N. Correia, and N. Guimaraes. “Hypervideo on the Web: models and techniques for video integration”. In: *International Journal of Computers & Applications* 23.2 (2001), pp. 90–98. DOI: 10455/6319.

[CZF04] T. Chambel, C. Zahn, and M. Finke. “Hypervideo design and support for contextualized learning”. In: *Proceedings of the IEEE International Conference on Advanced Learning Technologies, 2004*. IEEE, 2004, pp. 345–349. DOI: 10.1109/ICALT.2004.1357433.

[CCN11] T. Chambel, M. N. Chhaganlal, and L. A. R. Neng. “Towards immersive interactive video through 360&Deg; hypervideo”. In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology. ACE ’11*. New York, New York, USA: ACM, 2011, 78:1–78:2. DOI: 10.1145/2071423.2071518.

[CGa99] T. Chambel and N. Guimarães. “The role of hypervideo in learning environments”. In: *Proceedings of WebNet World Conference on the WWW and Internet 1999*. 31. Website (accessed April 26, 2014). Honolulu, Hawaii, USA: AACE, 1999, p. 1672. URL: http://www.editlib.org/p/7587.

[CGa02] T. Chambel and N. Guimarães. “Context perception in video-based hypermedia spaces”. In: *Proceedings of the 13th ACM Conference on Hypertext and Hypermedia. HYPERTEXT ’02*. New York, NY, USA: ACM, 2002, pp. 85–94. DOI: 10.1145/513338.513365.

[Cha05] A. Y. Chang. “Design of consistent SMIL documents for distributed multimedia presentation using temporal algebra”. In: *11th International Conference on Parallel and Distributed Systems, 2005 (ICPADS’05) - Vol. 1*. Vol. 1. IEEE, 2005, pp. 189–195. DOI: 10.1109/ICPADS.2005.128.

[CHS07] H.-B. Chang, H.-H. Hsu, and T. K. Shih. “Using interactive video technology for the development of game-based learning”. In: *International Conference on Parallel Processing Workshops, 2007. ICPPW 2007*. IEEE, 2007, pp. 24–28. DOI: 10.1109/ICPPW.2007.81.

[Cha+04] H.-B. Chang, H. H. Hsu, Y. C. Liao, T. K. Shih, and C.-T. Tang. “An object-based hypervideo authoring system”. In: *IEEE International Conference on Multimedia and Expo, 2004. ICME ’04*. Vol. 3. IEEE, 2004, pp. 2219–2222. DOI: 10.1109/ICME.2004.1394711.

[CHC08] H.-b. Chang, H.-h. Hsu, and L. R. Chao. “Interactive video game platform for game-based learning”. In: *Advances in Web Based Learning - ICWL 2008*. Ed. by F. Li, J. Zhao, T. Shih, R. Lau, Q. Li, and D. McLeod. Vol. 5145. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 232–240. DOI: 10.1007/978-3-540-85033-5_23.

[Cha08] D. K. Chaturvedi. *Soft computing: techniques and its applications in electrical engineering*. illustrated. Volume 103 of Studies in Computational Intelligence. Springer, 2008.

[CH97] J.-J. Chen and H.-M. Hang. “Source model for transform video coder and its application. II. Variable frame rate coding”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 7.2 (1997), pp. 299–311. DOI: 10.1109/76.564109.

[Che+09a] J.-H. Chen, T.-H. Wang, W.-C. Chang, L. R. Chao, and T. K. Shih. “Developing an interactive video game-based learning environment”. In: *Journal of Software* 4.2 (2009), pp. 132–139. DOI: 10.4304/jsw.4.2.132-139.

[Che+08] L. Chen, G.-C. Chen, C.-Z. Xu, J. March, and S. Benford. “EmoPlayer: A media player for video clips with affective annotations”. In: *Interacting with Computers* 20.1 (2008), pp. 17–28. DOI: 10.1016/j.intcom.2007.06.003.

[Che+02] S.-C. Chen, S.-T. Li, M.-L. Shyu, C. Zhan, and C. Zhang. “A multimedia semantic model for RTSP-based multimedia presentation systems”. In: *Proceedings. 4th International Symposium on Multimedia Software Engineering, 2002*. IEEE, 2002, pp. 124–131. DOI: 10.1109/MMSE.2002.1181604.

[Che+09b] Z. Chen, M. Zhang, L. Sun, and S. Yang. “Delay-guaranteed interactive multiview video streaming”. In: *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*. IEEE, 2009, pp. 1795–1798. DOI: 10.1109/ISCAS.2009.5118125.

[CK00] K. Cheng and Y. Kambayashi. “LRU-SP: a size-adjusted and popularity-aware LRU replacement algorithm for web caching”. In: *The 24th Annual International Computer Software and Applications Conference, 2000. COMPSAC 2000*. IEEE, 2000, pp. 48–53. DOI: 10.1109/CMPSAC.2000.884690.

[Che98] L. Cherkasova. *Improving WWW proxies performance with greedy-dual-size-frequency caching policy*. HPL-98-69R1. HP Laboratories Report. Website (accessed January 1, 2014). HP Laboratories, 1998. URL: http://www.hpl.hp.com/techreports/98/HPL-98-69R1.pdf.

[Che+09c] T. Cherrett, G. Wills, J. Price, S. Maynard, and I. E. Dror. “Making training more cognitively effective: making videos interactive”. In: *British Journal of Educational Technology* 40.6 (2009), pp. 1124–1134. DOI: 10.1111/j.1467-8535.2009.00985.x.

[COC11] G. Cheung, A. Ortega, and N.-M. Cheung. “Interactive streaming of stored multiview video using redundant frame structures”. In: *IEEE Transactions on Image Processing* 20.3 (2011), pp. 744–761. DOI: 10.1109/TIP.2010.2070074.

[CP02] S. Y. Cheung and C. S. Pencea. “BSFQ: bin sort fair queueing”. In: *IEEE INFOCOM 2002. 21st Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3. IEEE, 2002, pp. 1640–1649. DOI: 10.1109/INFCOM.2002.1019417.

[Chi+10] N. Chilamkurti, S. Zeadally, R. Soni, and G. Giambene. “Wireless multimedia delivery over 802.11e with cross-layer optimization techniques”. In: *Multimedia Tools and Applications* 47.1 (2010), pp. 189–205. DOI: 10.1007/s11042-009-0413-6.

[Chi+00] T.-c. Chiueh, T. Mitra, A. Neogi, and C.-K. Yang. “Zodiac: a history-based interactive video authoring system”. In: *Multimedia Systems* 8.3 (2000), pp. 201–211. DOI: 10.1007/s005300000045.

[Cho+03] K. Cho, Y. Ryu, Y. Won, and K. Koh. “Virtual interval caching scheme for interactive multimedia streaming workload”. In: *Computer and Information Sciences - ISCIS 2003*. Ed. by A. Yazici and C. Sener. Vol. 2869. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 276–283. DOI: 10.1007/978-3-540-39737-3_35.

[CI06] C.-O. Chow and H. Ishii. “A simulation study on multipoint-to-point video streaming over mobile ad hoc networks”. In: *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2006, pp. 1–5. DOI: 10.1109/PIMRC.2006.254176.

[CS97] C.-M. Chung and T. K. Shih. “On automatic generation of multimedia presentations”. In: *Information Sciences* 97.3-4 (1997), pp. 293–321. DOI: 10.1016/S0020-0255(96)00195-8.

[Chu+95] C.-M. Chung, T. K. Shih, J.-Y. Huang, Y.-H. Wang, and T.-F. Kuo. “An object-oriented approach and system for intelligent multimedia presentation designs”. In: *Proceedings of the International Conference on Multimedia Computing and Systems, 1995*. Washington, DC, USA: IEEE, 1995, pp. 278–281. DOI: 10.1109/MMCS.1995.484934.

[CP05] S. M. Chung and A. L. Pereira. “Timed Petri net representation of SMIL”. In: *IEEE Multimedia* 12.1 (2005), pp. 64–72. DOI: 10.1109/MMUL.2005.14.

[Cis14] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018*. White Paper. Website (accessed April 26, 2014). 2014. URL: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf.

[Cle12] Clear-Media. *ConciseClick: clickable video made easy*. Website (accessed April 26, 2014). 2012. URL: http://conciseclick.com/.

[Cli12] Clickable Video. *CLICKABLE VIDEO. Add a call to action to your video*. Website (accessed April 26, 2014). 2012. URL: http://www.clickablevideo.eu/.

[CES71] E. G. Coffman, M. Elphick, and A. Shoshani. “System deadlocks”. In: *ACM Computing Surveys* 3.2 (1971), pp. 67–78. DOI: 10.1145/356586.356588.

[CKR98] E. Cohen, B. Krishnamurthy, and J. Rexford. “Evaluating server-assisted cache replacement in the Web”. In: *Proceedings of the 6th Annual European Symposium on Algorithms. ESA'98*. Website (accessed April 26, 2014). London, UK, UK: Springer-Verlag, 1998, pp. 307–319. URL: http://www.cs.princeton.edu/~jrex/papers/esa98.pdf.

[Cor+04] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Algorithmen: eine Einführung*. Oldenbourg Wissenschaftsverlag GmbH, 2004.

[CC99] N. Correia and T. Chambel. “Active video watching using annotation”. In: *Proceedings of the 7th ACM International Conference on Multimedia (Part 2). MULTIMEDIA '99*. Vol. 99. New York, New York, USA: ACM, 1999, pp. 151–154. DOI: 10.1145/319878.319919.

[Cra12] C. Crawford. *Chris Crawford on Interactive Storytelling*. Pearson Education, 2012.

[CM99] I. F. Cruz and P. S. Mahalley. “Temporal synchronization in multimedia presentations”. In: *IEEE International Conference on Multimedia Computing and Systems, 1999*. Vol. 2. IEEE, 1999, pp. 851–856. DOI: 10.1109/MMCS.1999.778598.

[CL97] I. F. Cruz and W. T. Lucas. “Delaunay MM: a visual framework for multimedia presentation”. In: *1997 IEEE Symposium on Visual Languages, 1997*. IEEE, 1997, pp. 212–219. DOI: 10.1109/VL.1997.626585.

[Cut+09] S. Cutts, P. Davies, D. Newell, and N. Rowe. “Requirements for an adaptive multimedia presentation system with contextual supplemental support media”. In: *1st International Conference on Advances in Multimedia, 2009. MMEDIA '09*. IEEE, 2009, pp. 62–67. DOI: 10.1109/MMEDIA.2009.19.

[DTL06] X. Dai, S. Tabirca, and E. Lenihan. “JAS - an e-learning tool for building multimedia presentations”. In: *1st International Multi-Symposiums on Computer and Computational Sciences, 2006. IMSCCS'06*. Vol. 1. IEEE, 2006, pp. 743–746. DOI: 10.1109/IMSCCS.2006.81.

[DSS94] A. Dan, D. Sitaram, and P. Shahabuddin. “Scheduling policies for an on-demand video server with batching”. In: *Proceedings of the 2nd ACM International Conference on Multimedia. MULTIMEDIA ’94*. New York, NY, USA: ACM, 1994, pp. 15–23. DOI: 10.1145/192593.192614.

[DSP91] G. Davenport, T. A. Smith, and N. Pincever. “Cinematic primitives for multimedia”. In: *IEEE Computer Graphics and Applications* 11.4 (1991), pp. 67–74. DOI: 10.1109/38.126883.

[DB11] R. I. Davis and A. Burns. “A survey of hard real-time scheduling for multiprocessor systems”. In: *ACM Computing Surveys* 43.4 (2011), 35:1–35:44. DOI: 10.1145/1978802.1978814.

[DVL09] D. De Vleeschauwer and K. Laevens. “Performance of caching algorithms for IPTV on-demand services”. In: *IEEE Transactions on Broadcasting* 55.2 (2009), pp. 491–501. DOI: 10.1109/TBC.2009.2015983.

[DR06] R. Deltour and C. Roisin. “The Limsee3 multimedia authoring model”. In: *Proceedings of the 2006 ACM Symposium on Document Engineering. DocEng ’06*. New York, NY, USA: ACM, 2006, pp. 173–175. DOI: 10.1145/1166160.1166203.

[DKS89] A. Demers, S. Keshav, and S. Shenker. “Analysis and simulation of a fair queueing algorithm”. In: *SIGCOMM Computer Communication Review* 19.4 (1989), pp. 1–12. DOI: 10.1145/75246.75248.

[Den+02a] L. Y. Deng, R.-X. Chen, R.-C. Chang, and T.-S. Huang. “Adaptive content model for multimedia presentation”. In: *1st International Symposium on Cyber Worlds, 2002*. IEEE, 2002, pp. 209–216. DOI: 10.1109/CW.2002.1180881.

[Den+02b] L. Y. Deng, T. K. Shih, S.-H. Shiau, W.-C. Chang, and Y.-J. Liu. “Implementing a distributed lecture-on-demand multimedia presentation system”. In: *22nd International Conference on Distributed Computing Systems Workshops, 2002. Proceedings*. IEEE, 2002, pp. 111–115. DOI: 10.1109/ICDCSW.2002.1030756.

[Den12] A. Denk. “Eine Logging-Bibliothek für den SIVA-Player Konzeption, Implementierung und Auswertung”. Senior thesis. Passau, Germany: Universität Passau, 2012.

[Dij05] J. A. G. M. van Dijk. *The Network Society: Social Aspects of New Media*. SAGE Publications, 2005.

[Doh+03] J. Doherty, A. Girgensohn, J. Helfman, F. Shipman, and L. Wilcox. “Detail-on-demand hypervideo”. In: *Proceedings of the 11th ACM International Conference on Multimedia. MULTIMEDIA ’03*. New York, NY, USA: ACM, 2003, pp. 600–601. DOI: 10.1145/957135.957140.

[Dub12] Dublin Core Metadata Initiative Limited. *Dublin Core metadata element set, version 1.1*. Website (accessed April 26, 2014). 2012. URL: http://dublincore.org/documents/dces/.

[DR11] Z. A. Dwekat and G. N. Rouskas. “Worst-case fair bin sort queuing (WBSQ): an O(1) worst-case fair scheduler”. In: *2011 IEEE International Conference on Communications (ICC)*. 1. IEEE, 2011, pp. 1–5. DOI: 10.1109/icc.2011.5963218.

[EBU11] EBU (European Broadcasting Union), Department of Technology & Innovation. *P_META metadata library (EBU Tech 3295)*. Website (accessed April 26, 2014). 2011. URL: http://tech.ebu.ch/docs/tech/tech3295v2_2.pdf.

[ETS05] ETSI. *TV Anytime*. Website (accessed April 26, 2014). 2005. URL: http://www.etsi.org/technologies-clusters/technologies/broadcast/tv-anytime.

[Ech+98] M. Echiffre, C. Marchisio, P. Marchisio, P. Panicciari, and S. Del Rossi. "MHEG-5 - aims, concepts, and implementation issues". In: *IEEE Multimedia* 5.1 (1998), pp. 84–91. DOI: 10.1109/93.664745.

[Ecm13] Ecma International. *The JSON data interchange format*. Website (accessed April 26, 2014). 2013. URL: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.

[Egg+12] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler. "Waiting times in quality of experience for web based services". In: *Fourth International Workshop on Quality of Multimedia Experience (QoMEX), 2012*. IEEE, 2012, pp. 86–96. DOI: 10.1109/QoMEX.2012.6263888.

[Eli+09] S. Elias, L. Mathew, K. S. Easwarakumar, and R. Chbeir. "Automatic temporal formatting of multimedia presentations using dynamic petri nets". In: *Proceedings of 18th International Conference on Computer Communications and Networks, 2009. ICCCN 2009*. IEEE, 2009, pp. 1–6. DOI: 10.1109/ICCCN.2009.5235345.

[Emi+02] S. Emilda, L. Jacob, D. Ovidiu, and B. Prabhakaran. "Flexible disk scheduling for multimedia presentation servers". In: *10th IEEE International Conference on Networks, 2002. ICON 2002.* IEEE, 2002, pp. 151–155. DOI: 10.1109/ICON.2002.1033303.

[Emi+05] S. Emilda, L. Jacob, O. Daescu, and B. Prabhakaran. "Flexible strategies for disk scheduling in multimedia presentation servers". In: *Multimedia Tools and Applications* 26.1 (2005), pp. 81–99. DOI: 10.1007/s11042-005-6850-y.

[Erf93] R. Erfle. "Specification of temporal constraints in multimedia documents using HyTime". In: *Electronic Publishing* 6.4 (1993), pp. 397–411. URL: http://cajun.cs.nott.ac.uk/compsci/epo/papers/volume6/issue4/ep6x4rxe.pdf.

[Eun+94] S. Eun, E. S. No, H. C. Kim, H. Yoon, and S. R. Maeng. "Eventor: an authoring system for interactive multimedia applications". In: *Multimedia Systems* 2.3 (1994), pp. 129–140. DOI: 10.1007/BF01222125.

[Fei+99] Z. Fei, M. H. Ammar, I. Kamel, and S. Mukherjee. "Providing interactive functions through active client-buffer management in partitioned video multicast VoD systems". In: *Networked Group Communication*. Ed. by L. Rizzo and S. Fdida. Vol. 1736. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, pp. 152–169. DOI: 10.1007/978-3-540-46703-8_10.

[Fei+05] Z. Fei, M. H. Ammar, I. Kamel, and S. Mukherjee. “An active buffer management technique for providing interactive functions in broadcast video-on-demand systems”. In: *IEEE Transactions on Multimedia* 7.5 (2005), pp. 942–950. DOI: 10.1109/TMM.2005.854403.

[Fer+10] I. A. Fernandez, C. De Vleeschouwer, F. Lavigne, and X. Desurmont. “Worthy visual content on mobile through interactive video streaming”. In: *2010 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2010, pp. 412–417. DOI: 10.1109/ICME.2010.5583870.

[Fer+12] I. A. Fernandez, C. De Vleeschouwer, G. Toma, and L. Schumacher. “An interactive video streaming architecture featuring bitrate adaptation”. In: *Journal of Communications* 7.4 (2012), pp. 265–280. DOI: 10.4304/jcm.7.4.265-280.

[FV90] D. Ferrari and D. C. Verma. “A scheme for real-time channel establishment in wide-area networks”. In: *IEEE Journal on Selected Areas in Communications* 8.3 (1990), pp. 368–379. DOI: 10.1109/49.53013.

[Fer10] P. Ferreira. *MXF - a technical overview (EBU TECHNICAL REVIEW)*. Website (accessed April 29, 2013). 2010. URL: http://tech.ebu.ch/docs/techreview/trev_2010-Q3_MXF-2.pdf.

[Fic13] K. Fichtelmann. “Evaluation von Java Medienframeworks im Hinblick auf ihre Verwendbarkeit in Autorentools für interaktive nicht-lineare Videos”. Senior thesis. Passau, Germany: Universität Passau, 2013.

[FMF12] A. Field, J. Miles, and Z. Field. *Discovering Statistics Using R*. Reprinted 2013. SAGE Publications, 2012.

[FB04] M. Finke and D. Balfanz. “A reference architecture supporting hypervideo content for ITV and the internet domain”. In: *Computers & Graphics* 28.2 (2004), pp. 179–191. DOI: 10.1016/j.cag.2003.12.005.

[FR09] A. Fogarolli and M. Ronchetti. “Domain independent semantic representation of multimedia presentations”. In: *International Conference on Intelligent Networking and Collaborative Systems, 2009. INCOS ’09*. Washington, DC, USA: IEEE, 2009, pp. 31–38. DOI: 10.1109/INCOS.2009.80.

[FO01] N. L. S. Fonseca and R. M. Oliveira. “Role of download time as a key in Web cache management policies”. In: *IEEE Global Telecommunications Conference, 2001. GLOBECOM ’01*. Vol. 3. IEEE, 2001, pp. 2031–2035. DOI: 10.1109/GLOCOM.2001.965929.

[FHH00] A. P. Foong, Y. H. Hu, and D. M. Heisey. “Essence of an effective Web caching algorithm”. In: *Proceedings of the International Conference on Internet Computing*. Ed. by P. Graham and M. Maheswaran. CSREA Press, 2000, pp. 269–276.

[For89] R. Forsyth. *Expert systems: principles and case studies*. reprint. Chapman and Hall Computing. Series in Expert systems. Routledge, Chapman & Hall, Incorporated, 1989.

[For+10] R. Fortuna, L. A. Grieco, G. Boggia, and P. Camarda. “Quality adaptive end-to-end packet scheduling to avoid playout interruptions in Internet video streaming systems”. In: *Journal of Systems and Software* 83.8 (2010), pp. 1489 –1499. DOI: 10.1016/j.jss.2010.03.037.

[Fox89] E. A. Fox. “The coming revolution in interactive digital video”. In: *Communications of the ACM* 32.7 (1989), pp. 794–801. DOI: 10.1145/65445.65446.

[FCR13] A. L. Franzoni, C. P. Ceballos, and E. Rubio. “Interactive video enhanced learning-teaching process for digital native students”. In: *2013 IEEE 13th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2013, pp. 270–271. DOI: 10.1109/ICALT.2013.84.

[FHA00] M. Friedrich, S. Hollfelder, and K. Aberer. “Stochastic resource prediction and admission for interactive sessions on multimedia servers”. In: *Proceedings of the 8th ACM International Conference on Multimedia. MULTIMEDIA ’00*. New York, NY, USA: ACM, 2000, pp. 117–126. DOI: 10.1145/354384.354446.

[Fuj+91] K. Fujikawa, S. Shimojo, T. Matsuura, S. Nishio, and H. Miyahara. “Multimedia presentation system “Harmony” with temporal and active media”. In: *Proceedings of the Summer 1991 USENIX Conference*. Website (accessed April 26, 2014). 1991, pp. 75–94. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.3918&rep=rep1&type=pdf.

[Ful+10] R. Fulcher, C. Nesladek, J. Palmer, and C. Robertson. *Android UI design patterns*. Website/Presentation (accessed April 26, 2014). 2010. URL: http://dl.google.com/googleio/2010/android-android-ui-design-patterns.pdf.

[GB11] O. Gaggi and A. Bossi. “Analysis and verification of SMIL documents”. In: *Multimedia Systems* 17.6 (2011), pp. 487–506. DOI: 10.1007/s00530-011-0233-1.

[GC02] O. Gaggi and A. Celentano. “A visual authoring environment for prototyping multimedia presentations”. In: *4th International Symposium on Multimedia Software Engineering, 2002. Proceedings*. IEEE, 2002, pp. 206–213. DOI: 10.1109/MMSE.2002.1181614.

[GC05] O. Gaggi and A. Celentano. “Modelling synchronized hypermedia presentations”. In: *Multimedia Tools and Applications* 27.1 (2005), pp. 53–78. DOI: 10.1007/s11042-005-2714-8.

[Gao+11] B. Gao, J. Jansen, P. Cesar, and D. C. A. Bulterman. “Accurate and low-delay seeking within and across mash-ups of highly-compressed videos”. In: *Proceedings of the 21st International Workshop on Network and Operating Systems Support for Digital Audio and Video. NOSSDAV ’11*. New York, NY, USA: ACM, 2011, pp. 105–110. DOI: 10.1145/1989240.1989266.

[GAGM09] K. Geetha, N. Ammasai Gounden, and S. Monikandan. “SEMALRU: an implementation of modified web cache replacement algorithm”. In: *World Congress on Nature Biologically Inspired Computing, 2009. NaBIC 2009*. IEEE, 2009, pp. 1406–1410. DOI: 10.1109/NABIC.2009.5393711.

[GK00] D. Ghose and H. Kim. “Scheduling video streams in video-on-demand systems: a survey”. In: *Multimedia Tools and Applications* 11.2 (2000), pp. 167–195. DOI: 10.1023/A:1009681521536.

[GSW03] A. Girgensohn, F. Shipman, and L. Wilcox. “Hyper-Hitchcock: authoring interactive videos and generating interactive summaries”. In: *Proceedings of the 11th ACM International Conference on Multimedia. MULTIMEDIA ’03*. New York, NY, USA: ACM, 2003, pp. 92–93. DOI: 10.1145/957013.957030.

[Gir+04] A. Girgensohn, L. Wilcox, F. Shipman, and S. Bly. “Designing affordances for the navigation of detail-on-demand hypervideo”. In: *Proceedings of the Working Conference on Advanced Visual Interfaces. AVI ’04*. New York, New York, USA: ACM, 2004, pp. 290–297. DOI: 10.1145/989863.989913.

[Gol91] C. F. Goldfarb. “Standards-HyTime: a standard for structured hypermedia interchange”. In: *Computer* 24.8 (1991), pp. 81–84. DOI: 10.1109/2.84880.

[Gol94] S. J. Golestani. “A self-clocked fair queueing scheme for broadband applications”. In: *13th Proceedings IEEE INFOCOM ’94. Networking for Global Communications*. Vol. 2. IEEE, 1994, pp. 636–646. DOI: 10.1109/INFCOM.1994.337677.

[GZ98] S. Gollapudi and A. Zhang. “Buffer model and management in distributed multimedia presentation systems”. In: *Multimedia Systems* 6.3 (1998), pp. 206–218. DOI: 10.1007/s005300050089.

[GCCTC07] F. J. Gonzalez-Canete, E. Casilari, and A. Trivino-Cabrera. “Characterizing document types to evaluate Web cache replacement policies”. In: *4th European Conference on Universal Multiservice Networks, 2007. ECUMN ’07*. IEEE, 2007, pp. 3–11. DOI: 10.1109/ECUMN.2007.11.

[Got01] K. Gotthardt. *Grundlagen der Informationstechnik*. Einführungen - Informatik. LIT Verlag Münster, 2001.

[Got06] D. Gotz. “Scalable and adaptive streaming for non-linear media”. In: *Proceedings of the 14th Annual ACM International Conference on Multimedia. MULTIMEDIA ’06*. New York, NY, USA: ACM, 2006, pp. 357–366. DOI: 10.1145/1180639.1180717.

[GVC96] P. Goyal, H. M. Vin, and H. Chen. “Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks”. In: *ACM SIGCOMM Computer Communication Review* 26.4 (1996), pp. 157–168. DOI: 10.1145/248157.248171.

[GCD02] R. Grigoras, V. Charvillat, and M. Douze. “Optimizing hypervideo navigation using a Markov decision process approach”. In: *Proceedings of the 10th ACM International Conference on Multimedia. MULTIMEDIA ’02*. New York, NY, USA: ACM, 2002, pp. 39–48. DOI: 10.1145/641007.641014.

[Gri11] C. Grill. “Semantisches Zooming bei der Erstellung eines Szenengraphen für interaktive Videos (mit Fokus auf Usability und User-Centered Design)”. Master’s thesis. Passau, Germany: Universität Passau, 2011.

[Gul07] R. R. Gulati. *Monochrome and colour television*. reprint, revised. New Age International, 2007.

[Hak09] V. Hakkoymaz. “A specification model for temporal events in multimedia presentations”. In: *First International Conference on Networked Digital Technologies, 2009. NDT ’09*. IEEE, 2009, pp. 417–422. DOI: 10.1109/NDT.2009.5272109.

[HKO99] V. Hakkoymaz, J. Kraft, and G. Ozsoyoglu. “Constraint-based automation of multimedia presentation assembly”. In: *Multimedia Systems* 7.6 (1999), pp. 500–518. DOI: 10.1007/s005300050150.

[HS94] F. Halasz and M. Schwartz. “The Dexter hypertext reference model”. In: *Communications of the ACM* 37.2 (1994). Ed. by K. Grønbæk and R. H. Trigg, pp. 30–39. DOI: 10.1145/175235.175237.

[Hal+11] S. Halawa, D. Pang, N.-m. Cheung, and B. Girod. “ClassX: an open source interactive lecture streaming system”. In: *Proceedings of the 19th ACM International Conference on Multimedia. MM ’11*. New York, NY, USA: ACM, 2011, pp. 719–722. DOI: 10.1145/2072298.2072428.

[Ham06] R. I. Hammoud, ed. *Interactive video*. Signals and Communication Technology. Springer Berlin Heidelberg, 2006.

[Han11] A. Hanelly. *101 online video stats to make your eyes glaze over*. Website (accessed April 26, 2014). 2011. URL: http://www.business2community.com/online-marketing/101-online-video-stats-to-make-your-eyes-glaze-over-023074#!F3rPI.

[HB97] L. Hardman and D. C. A. Bulterman. “Document model issues for hypermedia”. In: *Handbook of Multimedia Information Management*. Ed. by W. I. Grosky, R. Jain, and R. Mehrotra. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997, pp. 39–68.

[HBR94] L. Hardman, D. C. A. Bulterman, and G. van Rossum. “The Amsterdam hypermedia model: adding time and context to the Dexter model”. In: *Communications of the ACM* 37.2 (1994), pp. 50–62. DOI: 10.1145/175235.175239.

[HWB97] L. Hardman, M. Worring, and D. C. A. Bulterman. “Integrating the Amsterdam hypermedia model with the standard reference model for intelligent multimedia presentation systems”. In: *Computer Standards & Interfaces* 18.6-7 (1997), pp. 497–507. DOI: 10.1016/S0920-5489(97)00014-7.

[HS95] R. L. Haskin and F. L. Stein. “A system for the delivery of interactive television programming”. In: *Compcon ’95. ’Technologies for the Information Superhighway’, Digest of Papers*. Washington, DC, USA: IEEE, 1995, pp. 209–215. DOI: 10.1109/CMPCON.1995.512388.

[HKS07] B. S. Hasler, B. Kersten, and J. Sweller. “Learner control, cognitive load and instructional animation”. In: *Applied Cognitive Psychology* 21.6 (2007), pp. 713–729. DOI: 10.1002/acp.1345.

[Hau08] M. Hausenblas. “Non-linear interactive media productions”. In: *Multimedia Systems* 14.6 (2008), pp. 405–413. DOI: 10.1007/s00530-008-0131-3.

[HRL96a] I. Herman, G. J. Reynolds, and J. van Loo. “Premo: an emerging standard for multimedia presentation part I: overview and framework”. In: *IEEE Multimedia* 3.3 (1996), pp. 83–89. DOI: 10.1109/MMUL.1996.556543.

[HRL96b] I. Herman, G. J. Reynolds, and J. van Loo. “Premo: an emerging standard for multimedia presentation part II: specification and applications”. In: *IEEE Multimedia* 3.4 (1996), pp. 72–75. DOI: 10.1109/93.556462.

[HVL01] R. Hjelsvold, S. Vdaygiri, and Y. Léauté. “Web-based personalization and management of interactive video”. In: *Proceedings of the 10th International Conference on World Wide Web. WWW ’01*. New York, NY, USA: ACM, 2001, pp. 129–139. DOI: 10.1145/371920.371969.

[HSA89] M. E. Hodges, R. M. Sasnett, and M. S. Ackerman. “A construction set for multimedia applications”. In: *IEEE Software* 6.1 (1989), pp. 37–43. DOI: 10.1109/52.16900.

[Hof12] J. Hoffmann. “Konzeption und Implementierung eines modularen Simulationsframeworks für interaktive Videos”. Master’s thesis. Passau, Germany: Universität Passau, 2012.

[HH06] P. Hoffmann and M. Herczeg. “Hypervideo vs. storytelling integrating narrative intelligence into hypervideo”. In: *Technologies for Interactive Digital Storytelling and Entertainment*. Ed. by S. Göbel, R. Malkewitz, and I. Iurgel. Vol. 4326. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 37–48. DOI: 10.1007/11944577_4.

[HKH08] P. Hoffmann, T. Kochems, and M. Herczeg. “HyLive: hypervideo-authoring for live television”. In: *Changing Television Environments*. Ed. by M. Tscheligi, M. Obrist, and A. Lugmayr. Vol. 5066. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 51–60. DOI: 10.1007/978-3-540-69478-6_6.

[HHF09] C. Hofmann, N. Hollender, and D. W. Fellner. “Task- and process-related design of video annotation systems”. In: *Mensch & Computer 2009: Grenzenlos frei!?* Ed. by H. Wandke, S. Kain, and D. Struve. Website (accessed April 26, 2014). München, Germany: Oldenbourg Verlag, 2009, pp. 173–182. URL: http://dl.mensch-und-computer.de/handle/123456789/326.

[HFP99] T. Hollerer, S. Feiner, and J. Pavlik. “Situated documentaries: embedding multimedia presentations in the real world”. In: *The 3rd IEEE International Symposium on Wearable Computers, 1999. Digest of Papers*. Washington, DC, USA: IEEE, 1999, pp. 79–86. DOI: 10.1109/ISWC.1999.806664.

[HA98] S. Hollfelder and K. Aberer. “An admission control framework for applications with variable consumption rates in client-pull architectures”. In: *Advances in Multimedia Information Systems*. Vol. 1508. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1998, pp. 82–97. DOI: 10.1007/3-540-49651-3_9.

[HA99] S. Hollfelder and K. Aberer. “Resource prediction for admission control of interactive multimedia sessions”. In: *Database Semantics*. Ed. by R. Meersman, Z. Tari, and S. Stevens. Vol. 11. IFIP - The International Federation for Information Processing. Springer US, 1999, pp. 27–46. DOI: 10.1007/978-0-387-35561-0_4.

[Hon13] Honkytonk Films. *Klynt*. Website (accessed April 26, 2014). 2013. URL: http://www.klynt.net/.

[Hos+11] T. Hossfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz. “Quantification of YouTube QoE via Crowdsourcing”. In: *IEEE International Symposium on Multimedia (ISM), 2011*. 2011, pp. 494–499. DOI: 10.1109/ISM.2011.87.

[Hos+12] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen. “Initial delay vs. interruptions: between the devil and the deep blue sea”. In: *Fourth International Workshop on Quality of Multimedia Experience (QoMEX), 2012*. IEEE, 2012, pp. 1–6. DOI: 10.1109/QoMEX.2012.6263849.

[Hsu+05] H.-H. Hsu, T. Shih, H.-B. Chang, Y.-C. Liao, and C.-T. Tang. “Hyperinteractive video browsing by a remote controller and hand gestures”. In: *Embedded and Ubiquitous Computing - EUC 2005 Workshops*. Ed. by T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, and L. T. Yang. Vol. 3823. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 547–555. DOI: 10.1007/11596042_57.

[HF06] J. Hu and L. Feijs. “IPML: extending SMIL for distributed multimedia presentations”. In: *Interactive Technologies and Sociotechnical Systems*. Ed. by H. Zha, Z. Pan, H. Thwaites, A. C. Addison, and M. Forte. Vol. 4270. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 60–70. DOI: 10.1007/11890881_8.

[HL06] X.-S. Hua and S. Li. “Interactive video authoring and sharing based on two-layer templates”. In: *Proceedings of the 1st ACM International Workshop on Human-Centered Multimedia. HCM ’06*. New York, NY, USA: ACM, 2006, pp. 65–74. DOI: 10.1145/1178745.1178758.

[HW98] C.-M. Huang and C. Wang. “Synchronization for interactive multimedia presentations”. In: *IEEE Multimedia* 5.4 (1998), pp. 44–62. DOI: 10.1109/93.735868.

[Hun97] Y.-C. Hung. “A hypervideo system generator”. In: *Software: Practice and Experience* 27.11 (1997), pp. 1263–1281. DOI: 10.1002/(SICI)1097-024X(199711)27:11<1263::AID-SPE129>3.0.CO;2-3.

[IBM11] IBM Institute for Business Value. *Digital consumer survey 2011*. Whitepaper (accessed April 26, 2014). 2011. URL: http://public.dhe.ibm.com/common/ssi/ecm/en/gbf03032usen/GBF03032USEN.PDF.

[IEE02] IEEE Learning Technology Standards Committee (LTSC). *Learning Object Metadata (LOM), (1484.12.1-2002 IEEE Standard for Learning Object Metadata)*. Website (accessed April 26, 2014). 2002. URL: http://ltsc.ieee.org/wg12/.

[ISO92] ISO/EIC. *ISO/IEC 11072:1992 information technology – Computer graphics – Computer graphics reference model*. Standard/Website (accessed April 26, 2014). 1992. URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=19059.

[ISO09] ISO/IEC. *MPEG-7. ISO/IEC 15938. Multimedia content description interface*. Standard/Website (accessed April 26, 2014). 2009. URL: http://mpeg.chiariglione.org/standards/mpeg-7.

[Ill09] J. Illik. *Formale Methoden der Informatik: Von der Automatentheorie zu Algorithmen und Datenstrukturen*. Reihe Technik. Expert-Verlag, 2009.

[IW06] Incisive Interactive Marketing LLC and T. Wegert. *What ever happened to clickable video?* Website (accessed April 26, 2014). 2006. URL: http://www.clickz.com/clickz/column/1704276/what-ever-happened-clickable-video.

[Inn11] Innoteams GmbH. *ADIVI add digital information to video*. Website (accessed April 26, 2014). 2011. URL: http://www.adivi.net/.

[Ino+10] M. Inoue, H. Kimata, K. Fukazawa, and N. Matsuura. "Interactive panoramic video streaming system over restricted bandwidth network". In: *Proceedings of the International Conference on Multimedia. MM '10*. New York, New York, USA: ACM, 2010, pp. 1191–1194. DOI: 10.1145/1873951.1874184.

[Int85] International Organization for Standardization. *https://www.iso.org/obp/ui/#iso:std:iso:5807:ed-1:v1:en*. Website (accessed April 26, 2014). 1985. URL: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=11955.

[Int11] International Telecommunication Union. *H.761: Nested context language (NCL) and Ginga-NCL*. Website (accessed April 26, 2014). 2011. URL: http://www.itu.int/rec/T-REC-H.761-201106-I/en.

[IA01] E. Ishikawa and C. L. Amorim. "Cooperative video caching for interactive and scalable VoD systems". In: *Networking - ICN 2001*. Vol. 2094. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 776–785. DOI: 10.1007/3-540-47734-9_77.

[Jae12] J. Jaeger. *Use cases - open hypervideo prototypes & implementations*. Website (accessed April 26, 2014). 2012. URL: http://www.open-hypervideo.org/use-cases.html.

[JW95] R. Jain and K. Wakimoto. "Multiple perspective interactive video". In: *Proceedings of the International Conference on Multimedia Computing and Systems, 1995*. 1995, pp. 202–211. DOI: 10.1109/MMCS.1995.484925.

[JPP00] A. Jalali, R. Padovani, and R. Pankaj. "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system". In: *2000 IEEE 51st Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo*. Vol. 3. IEEE, 2000, pp. 1854–1858. DOI: 10.1109/VETECS.2000.851593.

[Jan10] J.-O. Janda. “Durchführung und Auswertung von Performancetests an XML Daten der SIVA Suite”. Thesis. Passau, Germany: Universität Passau, 2010.

[JB08] J. Jansen and D. C. A. Bulterman. “Enabling adaptive time-based web applications with SMIL state”. In: *Proceeding of the 8th ACM Symposium on Document Engineering. DocEng '08*. New York, NY, USA: ACM, 2008, pp. 18–27. DOI: 10.1145/1410140.1410146.

[JB09] J. Jansen and D. C. A. Bulterman. “SMIL State: an architecture and implementation for adaptive time-based web applications”. In: *Multimedia Tools and Applications* 43.3 (2009), pp. 203–224. DOI: 10.1007/s11042-009-0270-3.

[JCB10] J. Jansen, P. Cesar, and D. C. A. Bulterman. “A model for editing operations on active temporal multimedia documents”. In: *Proceedings of the 10th ACM Symposium on Document Engineering. DocEng '10*. New York, NY, USA: ACM, 2010, pp. 87–96. DOI: 10.1145/1860559.1860579.

[JE98] H. Jiang and A. K. Elmagarmid. “Spatial and temporal content-based access to hypervideo databases”. In: *The VLDB Journal* 7.4 (1998), pp. 226–238. DOI: 10.1007/s007780050066.

[JB01] S. Jin and A. Bestavros. “GreedyDual* Web caching algorithm: exploiting the two sources of temporal locality in web request streams”. In: *Computer Communications* 24.2 (2001), pp. 174–183. DOI: 10.1016/S0140-3664(00)00312-1.

[JLK08] T. Jokela, J. T. Lehikoinen, and H. Korhonen. “Mobile multimedia presentation editor: enabling creation of audio-visual stories on mobile devices”. In: *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems. CHI 08*. New York, NY, USA: ACM, 2008, pp. 63–72. DOI: 10.1145/1357054.1357066.

[Jou+98] M. Jourdan, N. Layaïda, C. Roisin, L. Sabry-Ismaïl, and L. Tardif. “Madeus, an authoring environment for interactive multimedia documents”. In: *Proceedings of the 6th ACM International Conference on Multimedia. MULTIMEDIA '98*. New York, NY, USA: ACM, 1998, pp. 267–272. DOI: 10.1145/290747.290780.

[KU99] M. Kaji and K. Uehara. “Creating multimedia presentation based on constraint satisfaction problems in multimedia databases”. In: *1999 International Symposium on Database Applications in Non-Traditional Environments, 1999. (DANTE '99) Proceedings.* IEEE, 1999, pp. 143–151. DOI: 10.1109/DANTE.1999.844952.

[KTM07] T. H. Kaskalis, T. D. Tzidamis, and K. Margaritis. “Multimedia authoring tools: the quest for an educational package”. In: *Educational Technology & Society* 10.3 (2007). Website (accessed April 26, 2014), pp. 135–162. URL: http://www.ifets.info/journals/10_3/10.pdf.

[Kat+96] A. Katkere, J. Schlenzig, A. Gupta, and R. Jain. “Interactive video on WWW: beyond VCR-like interfaces”. In: *Computer Networks and ISDN Systems* 28.7-11 (1996), pp. 1559–1572. DOI: 10.1016/0169-7552(96)00025-6.

[Ke+06] C.-H. Ke, C.-H. Lin, C.-K. Shieh, and W.-S. Hwang. “A novel realistic simulation tool for video transmission over wireless network”. In: *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. Vol. 1. IEEE, 2006, pp. 275–283. DOI: 10.1109/SUTC.2006.1636186.

[Kel+95] P. H. Kelly, A. Katkere, D. Y. Kuramura, S. Moezzi, and S. Chatterjee. “An architecture for multiple perspective interactive video”. In: *Proceedings of the 3rd ACM International Conference on Multimedia. MULTIMEDIA ’95*. New York, NY, USA: ACM, 1995, pp. 201–212. DOI: 10.1145/217279.215267.

[KD97] C. Keramane and A. Duda. “Operator based composition of structured multimedia presentations”. In: *From Multimedia Services to Network Services*. Lecture Notes in Computer Science 1356 (1997). Ed. by A. Danthine and C. Diot. Website (accessed April 26, 2014), pp. 1–32. URL: http://www.springerlink.com/index/D5568R436290594M.pdf.

[KRO11] N. Q. M. Khiem, G. Ravindra, and W. T. Ooi. “Towards understanding user tolerance to network latency in zoomable video streaming”. In: *Proceedings of the 19th ACM International Conference on Multimedia. MM ’11*. New York, NY, USA: ACM, 2011, pp. 977–980. DOI: 10.1145/2072298.2071917.

[Kim+00] J. W. Kim, Y.-G. Kim, H. Song, T.-Y. Kuo, Y. J. Chung, and C.-C. J. Kuo. “TCP-friendly Internet video streaming employing variable frame-rate encoding and interpolation”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 10.7 (2000), pp. 1164–1177. DOI: 10.1109/76.875520.

[Kim+11] J. Kim, D. Hwang, K. Kim, C. Jung, and W. Kim. “Development of LMS/LCMS (contents link module) real-time interactive in videos for maximizing the effect of learning”. In: *Web Information Systems and Mining*. Vol. 6988. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 444–451. DOI: 10.1007/978-3-642-23982-3_54.

[KK12] Y.-J. Kim and J. Kim. “ARC-H: adaptive replacement cache management for heterogenous storage devices”. In: *Journal of Systems Architecture* 58.2 (2012), pp. 86–97. DOI: 10.1016/j.sysarc.2011.12.002.

[KST04] P. King, P. Schmitz, and S. Thompson. “Behavioral reactivity and real time programming in XML: functional programming meets SMIL animation”. In: *Proceedings of the 2004 ACM Symposium on Document Engineering. DocEng ’04*. New York, NY, USA: ACM, 2004, pp. 57–66. DOI: 10.1145/1030397.1030411.

[Kip01] M. Kipp. “ANVIL - a generic annotation tool for multimodal dialogue”. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*. Website (accessed April 26, 2014). 2001, pp. 1367–1370. URL: http://www.dfki.de/~kipp/public_archive/kipp2001-eurospeech.pdf.

[KRW03] J. Klaue, B. Rathke, and A. Wolisz. “EvalVid - a framework for video transmission and quality evaluation”. In: *Computer Performance Evaluation. Modelling Techniques and Tools*. Ed. by P. Kemper and W. H. Sanders. Vol. 2794. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 255–272. DOI: 10.1007/978-3-540-45232-4_16.

[KHM08] T. Kleinberger, A. Holzinger, and P. Müller. “Adaptive multimedia presentations enabling universal access in technology enhanced situational learning”. In: *Universal Access in the Information Society* 7.4 (2008), pp. 223–245. DOI: 10.1007/s10209-008-0122-3.

[Kli13] Klickable Inc. *Klickable beta*. Website (accessed April 26, 2014). 2013. URL: http://www.klickable.tv/.

[KTP09] A. Konstantinidis, T. Tsiatsos, and A. Pomportsis. “Collaborative virtual learning environments: design and evaluation”. In: *Multimedia Tools and Applications* 44.2 (2009), pp. 279–304. DOI: 10.1007/s11042-009-0289-5.

[Kos+04] H. Kosch, A. Mostefaoui, L. Böszörmenyi, and L. Brunie. “Heuristics for optimizing multi-clip queries in video databases”. In: *Multimedia Tools and Applications* 22.3 (2004), pp. 235–262. DOI: 10.1023/B:MTAP.0000017030.45487.43.

[KÏ11] J. Köstler. “Wiedergabe interaktiver Videos auf Android-Smartphones - Technologieevaluierung, Konzeption und Implementierung”. Senior thesis. Passau, Germany: Universität Passau, 2011.

[KWW96] M. Kozuch, W. Wolf, and A. Wolfe. “New challenges for video servers: performance of non-linear applications under user choice”. In: *1996 IEEE International Conference on Computer Design: VLSI in Computers and Processors, 1996. ICCD ’96. Proceedings*. Austin, TX, USA, 1996, pp. 145–146. DOI: 10.1109/ICCD.1996.563549.

[KWW00] M. Kozuch, W. Wolf, and A. Wolfe. “An experimental analysis of digital video library servers”. In: *Multimedia Systems* 8.2 (2000), pp. 135–145. DOI: 10.1007/s005300050156.

[KS12] S. S. Krishnan and R. K. Sitaraman. “Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs”. In: *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*. IMC ’12. New York, NY, USA: ACM, 2012, pp. 211–224. DOI: 10.1145/2398776.2398799.

[Kri+11] D. K. Krishnappa, S. Khemmarat, L. Gao, and M. Zink. “On the feasibility of prefetching and caching for online TV services: a measurement study on hulu”. In: *Proceedings of the 12th International Conference on Passive and Active Measurement. PAM ’11*. Springer-Verlag Berlin Heidelberg, 2011, pp. 72–80. DOI: 10.1007/978-3-642-19260-9_8.

[Kru52] W. H. Kruskal. “A Nonparametric test for the Several Sample Problem”. In: *The Annals of Mathematical Statistics* 23.4 (Dec. 1952), pp. 525–540. DOI: 10.1214/aoms/1177729332.

[KW52] W. H. Kruskal and W. A. Wallis. “Use of Ranks in One-Criterion Variance Analysis”. In: *Journal of the American Statistical Association* 47.260 (1952), pp. 583–621. DOI: 10.1080/01621459.1952.10483441.

[Kuc13] M. Kuchler. “Interaktive nicht-lineare Videos - Anwendungsszenarien und benutzerfreundliche Gestaltung eines Autorentools”. Master’s thesis. Passau, Germany: Universität Passau, 2013.

[KCT07] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp. “Client-driven selective streaming of multiview video for interactive 3DTV”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.11 (2007), pp. 1558–1565. DOI: 10.1109/TCSVT.2007.903664.

[LGaCB10] R. Laiola Guimarães, P. Cesar, and D. C. A. Bulterman. “Creating and sharing personalized time-based annotations of videos on the web”. In: *Proceedings of the 10th ACM Symposium on Document Engineering. DocEng ’10*. New York, New York, USA: ACM, 2010, pp. 27–36. DOI: 10.1145/1860559.1860567.

[LGaMDRCGS08] R. Laiola Guimarães, R. Monteiro De Resende Costa, and L. F. Gomes Soares. “Composer: authoring tool for iTV programs”. In: *Changing Television Environments*. Ed. by M. Tscheligi, M. Obrist, and A. Lugmayr. Vol. 5066. Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 2008, pp. 61–71. DOI: 10.1007/978-3-540-69478-6_7.

[Lan+04] L. Lancerica, L. Dairaine, F. De Belleville, H. Thalmensy, and C. Fraboul. “MITv - a solution for an interactive TV based on IP multicast over satellite”. In: *IEEE International Conference on Multimedia and Expo, 2004. ICME ’04.* Vol. 3. IEEE, 2004, pp. 2159–2162. DOI: 10.1109/ICME.2004.1394696.

[Lan11] A. Lang. “Wiedergabe interaktiver Videos auf dem iPhone: Konzeption und Implementierung unter Berücksichtigung von Methoden des Usability-Engineerings”. Master’s thesis. Passau, Germany: Universität Passau, 2011.

[LSC10] R. Laraspata, D. Striccoli, and P. Camarda. “A scheduling algorithm for interactive video streaming in UMTS networks”. In: *2010 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2010, pp. 997–1002. DOI: 10.1109/ISCC.2010.5546653.

[LSI96] N. Layaida and L. Sabry-Ismail. “Maintaining temporal consistency of multimedia documents using constraint networks”. In: *Proc. SPIE 2667, Multimedia Computing and Networking 1996*. SPIE Digital Library, 1996, pp. 124–135. DOI: 10.1117/12.235866.

[LSIR02] N. Layaida, L. Sabry-Ismail, and C. Roisin. “Dealing with uncertain durations in synchronized multimedia presentations”. In: *Multimedia Tools and Applications* 18.3 (2002), pp. 213–231. DOI: 10.1023/A:1019944800320.

[LP10] I. Lee and J. H. Park. “A scalable and adaptive video streaming framework over multiple paths”. In: *Multimedia Tools and Applications* 47.1 (2010), pp. 207–224. DOI: 10.1007/s11042-009-0414-5.

[LSC03] J. F. Lee, Y. Sun, and M. C. Chen. “On maximum rate control of weighted fair scheduling for transactional systems”. In: *24th IEEE Real-Time Systems Symposium, 2003. RTSS 2003*. IEEE, 2003, pp. 335–344. DOI: 10.1109/REAL.2003.1253279.

[LSY04] S.-H. Lee, D.-R. Shin, and H. Youn. “Weighted fair scheduling algorithm for QoS of input-queued switches”. In: *Network and Parallel Computing*. Ed. by H. Jin, G. R. Gao, Z. Xu, and H. Chen. Vol. 3222. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 366–373. DOI: 10.1007/978-3-540-30141-7_51.

[LC08] S. Lee and K. Chung. “Buffer-driven adaptive video streaming with TCP-friendliness”. In: *Computer Communications* 31.10 (2008), pp. 2621–2630. DOI: 10.1016/j.comcom.2008.02.011.

[LO98] T. Lee and G. Ozsoyoglu. “Query processing techniques for multimedia presentation graphs”. In: *Eighth International Workshop on Research Issues in Data Engineering, 1998. ’Continuous-Media Databases and Applications’. Proceedings*. IEEE, 1998, pp. 130–138. DOI: 10.1109/RIDE.1998.658287.

[Lee+99] T. Lee, L. Sheng, T. Bozkaya, N. H. Balkir, Z. Meral Ozsoyoglu, and Z. M. Ozsoyoglu. “Querying multimedia presentations based on content”. In: *IEEE Transactions on Knowledge and Data Engineering* 11.3 (1999), pp. 361–385. DOI: 10.1109/69.774099.

[Leh11] J. Lehtimaki. *Smartphone UI patterns*. Website (accessed April 26, 2014). 2011. URL: http://www.androiduipatterns.com/p/android-ui-pattern-collection.html.

[LC03] Y.-W. Leung and T. K. C. Chan. “Design of an interactive video-on-demand system”. In: *IEEE Transactions on Multimedia* 5.1 (2003), pp. 130–140. DOI: 10.1109/TMM.2003.808818.

[LBH09] T. Li, D. Baumberger, and S. Hahn. “Efficient and scalable multiprocessor fair scheduling using distributed weighted round-robin”. In: *Proceedings of the 14th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. PPoPP ’09*. New York, NY, USA: ACM, 2009, pp. 65–74. DOI: 10.1145/1504176.1504188.

[Li+96] V. K. Li, L. Wanjiun, X. Qiu, and E. W. M. Wong. “Performance model of interactive video-on-demand systems”. In: *IEEE Journal on Selected Areas in Communications* 14.6 (1996), pp. 1099–1109. DOI: 10.1109/49.508281.

[LO09] Y. Li and K. Ong. “Optimized scalable cache management for video streaming system”. In: *Multimedia Tools and Applications* 44.1 (2009), pp. 65–86. DOI: 10.1007/s11042-009-0264-1.

[LS11] Z. Li and G. Simon. “Time-shifted TV in content centric networks: the case for cooperative in-network caching”. In: *2011 IEEE International Conference on Communications (ICC)*. 2011, pp. 1–6. DOI: 10.1109/icc.2011.5963380.

[LL98] W. Liao and V. O. K. Li. “Synchronization of distributed multimedia systems with user interactions”. In: *Multimedia Systems* 6.3 (1998), pp. 196–205. DOI: 10.1007/s005300050088.

[LS02] W.-K. Liao and P.-H. Shih. “Architecture of proxy partial caching using HTTP for supporting interactive video and cache consistency”. In: *Proceedings. 11th International Conference on Computer Communications and Networks, 2002. Proceedings*. IEEE, 2002, pp. 216–222. DOI: 10.1109/ICCCN.2002.1043069.

[LK08] A. Lie and J. Klaue. “Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video”. In: *Multimedia Systems* 14.1 (2008), pp. 33–50. DOI: 10.1007/s00530-007-0110-0.

[Lie+05] G. Liebl, H. Jenkac, T. Stockhammer, and C. Buchner. “Joint buffer management and scheduling for wireless video streaming”. In: *Networking - ICN 2005*. Ed. by P. Lorenz and P. Dini. Vol. 3420. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 882–891. DOI: 10.1007/978-3-540-31956-6_104.

[Lig13] Lighthouse Websites, LLC. *Rich media applications*. Website (accessed April 26, 2014). 2013. URL: http://www.lmgit.com/services/rich-media-applications/.

[LHL13] F.-C. Lin, J.-S. Hong, and B. M. T. Lin. “Sequence optimization for media objects with due date constraints in multimedia presentations from digital libraries”. In: *Information Systems* 38.1 (2013), pp. 82–96. DOI: 10.1016/j.is.2012.05.008.

[LO96] J. Lin and Z. M. Ozsoyoglu. “Processing OODB queries by O-Algebra”. In: *Proceedings of the 5th International Conference on Information and Knowledge Management. CIKM ’96*. New York, NY, USA: ACM, 1996, pp. 134–142. DOI: 10.1145/238355.238470.

[Lin13] LinkedTV Consortium. *LinkedTV*. Website (accessed April 26, 2014). 2013. URL: http://www.linkedtv.eu/.

[LG90a] T. D. C. Little and A. Ghafoor. “Multimedia objects models for synchronization and databases”. In: *6th International Conference on Data Engineering, 1990. Proceedings*. IEEE, 1990, pp. 20–27. DOI: 10.1109/ICDE.1990.113450.

[LG90b] T. D. C. Little and A. Ghafoor. “Synchronization and storage models for multimedia objects”. In: *IEEE Journal on Selected Areas in Communications* 8.3 (1990), pp. 413–427. DOI: 10.1109/49.53017.

[LL04] J. Liu and B. Li. “A QoS-based joint scheduling and caching algorithm for multimedia objects”. In: *World Wide Web* 7.3 (2004), pp. 281–296. DOI: 10.1023/B:WWWJ.0000028181.13079.80.

[Liu+04] W. Liu, C. T. Chou, Z. Yang, and X. Du. “Popularity-wise proxy caching for interactive streaming media”. In: *29th Annual IEEE International Conference on Local Computer Networks, 2004*. IEEE, 2004, pp. 250–257. DOI: 10.1109/LCN.2004.96.

[Liu+10] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao. "RD-optimized interactive streaming of multiview video with multiple encodings". In: *Journal of Visual Communication and Image Representation* 21.5-6 (2010), pp. 523–532. DOI: 10.1016/j.jvcir.2010.02.004.

[LB06] A. Lo Bue. "Using layout profiles in MPEG-4 hypervideo presentation". In: *Eurographics Italian Chapter Conference*. Ed. by G. Gallo, S. Battiato, and F. Stanco. Website (accessed April 26, 2014). Eurographics, 2006, pp. 273–277. URL: http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.97.5567.

[Loh09] J. Lohr. *High definition media services: Zukunft des Rundfunks im Internet: Technologien und Anwendungsperspektiven*. Fachverlag Schiele & Schoen, 2009.

[LM12] M. Lombardi and M. Milano. "Optimal methods for resource allocation and scheduling: a cross-disciplinary survey". In: *Constraints* 17.1 (2012), pp. 51–85. DOI: 10.1007/s10601-011-9115-6.

[Lus+09] D. L. Lusk, A. D. Evans, T. R. Jeffrey, K. R. Palmer, C. S. Wikstrom, and P. E. Doolittle. "Multimedia learning and individual differences: mediating the effects of working memory capacity with segmentation". In: *British Journal of Educational Technology* 40.4 (2009), pp. 636–651. DOI: 10.1111/j.1467-8535.2008.00848.x.

[Mac91] R. MacNeil. "Generating multimedia presentations automatically using TYRO, the constraint, case-based designer's apprentice". In: *1991 IEEE Workshop on Visual Languages, 1991. Proceedings*. IEEE, 1991, pp. 74–79. DOI: 10.1109/WVL.1991.238847.

[MD89] W. E. Mackay and G. Davenport. "Virtual video editing in interactive multimedia applications". In: *Communications of the ACM* 32.7 (1989), pp. 802–810. DOI: 10.1145/65445.65447.

[Mak+10] M. Makar, A. Mavlankar, P. Agrawal, and B. Girod. "Real-time video streaming with interactive region-of-interest". In: *17th IEEE International Conference on Image Processing (ICIP), 2010*. IEEE, 2010, pp. 4437–4440. DOI: 10.1109/ICIP.2010.5653982.

[Mat12] MatchWare A/S. *Mediator 9 - multimedia authoring software*. Website (accessed April 26, 2014). 2012. URL: http://www.matchware.com/en/products/mediator/default.htm.

[MK91] K. F. Matta and G. M. Kern. "Interactive videodisc instruction: the influence of personality on learning". In: *International Journal of Man-Machine Studies* 35.4 (1991), pp. 541–552. DOI: 10.1016/S0020-7373(05)80091-4.

[Mat11] K. Matusik. "Erweiterte Videos - Begriffsabgrenzung, Möglichkeiten der Interaktion und Evaluation von Autorentools". Senior thesis. Passau, Germany: Universität Passau, 2011.

[MF11] T. Maugey and P. Frossard. "Interactive multiview video system with low decoding complexity". In: *18th IEEE International Conference on Image Processing (ICIP), 2011*. IEEE, 2011, pp. 589–592. DOI: 10.1109/ICIP.2011.6116618.

[MPG07] K. Mayer-Patel and D. Gotz. “Scalable, adaptive streaming for nonlinear media”. In: *IEEE Multimedia* 14.3 (2007), pp. 68–83. DOI: 10.1109/MMUL.2007.63.

[MR06] J. McMullan and I. Richardson. “The mobile phone: a hybrid multiplatform medium”. In: *Proceedings of the 3rd Australasian Conference on Interactive Entertainment. IE ’06*. Website (accessed April 26, 2014). Australia: Murdoch University, 2006, pp. 103–108. URL: http://dl.acm.org/citation.cfm?id=1231910.

[MA11] R. Mehmood and R. Alturki. “A scalable multimedia QoS architecture for ad hoc networks”. In: *Multimedia Tools and Applications* 54.3 (2011), pp. 551–568. DOI: 10.1007/s11042-010-0569-0.

[MM94] S. R. L. Meira and A. E. L. Moura. “A scripting language for multimedia presentations”. In: *Proceedings of IEEE International Conference on Multimedia Computing and Systems, 1994*. IEEE, 1994, pp. 484–489. DOI: 10.1109/MMCS.1994.292493.

[MGK11] B. Meixner, S. Gattermann, and H. Kosch. “Nichtlineares Multiple Choice Quiz zur Unterstützung unterschiedlicher Lernniveaus”. In: *Mensch & Computer 2011: überMEDIEN|ÜBERmorgen*. Ed. by M. Eibl. Website (accessed April 26, 2014). München: Oldenbourg Verlag, 2011, pp. 275–278. URL: http://dl.mensch-und-computer.de/handle/123456789/1523.

[MGK12] B. Meixner, C. Grill, and H. Kosch. “A zooming concept for an interactive non-linear video authoring software”. In: *Mensch & Computer 2012: interaktiv informiert - allgegenwärtig und allumfassend!?* Ed. by H. Reiterer and O. Deussen. Website (accessed April 26, 2014). München: Oldenbourg Verlag, 2012, pp. 283–292. URL: http://dl.mensch-und-computer.de/handle/123456789/2892.

[MH12] B. Meixner and J. Hoffmann. “Intelligent download and cache management for interactive non-linear video”. In: *Multimedia Tools and Applications* (2012), pp. 1–44. DOI: 10.1007/s11042-012-1158-1.

[MK12] B. Meixner and H. Kosch. “Interactive non-linear video: definition and XML structure”. In: *Proceedings of the 2012 ACM Symposium on Document Engineering. DocEng ’12*. New York, NY, USA: ACM, 2012, pp. 49–58. DOI: 10.1145/2361354.2361367.

[MK13] B. Meixner and H. Kosch. “Creating and presenting interactive non-linear video stories with the SIVA Suite”. In: *Adjunct Proceedings of the 1st International Workshop on Interactive Content Consumption at EuroITV*. Website (accessed April 26, 2014). 2013, pp. 160–165. URL: http://www.fascinate-project.eu/index.php/wsicceuroitv2013/.

[MKK11] B. Meixner, J. Köstler, and H. Kosch. “A mobile player for interactive non-linear video”. In: *Proceedings of the 19th ACM International Conference on Multimedia. MM ’11*. New York, New York, USA: ACM, 2011, pp. 779–780. DOI: 10.1145/2072298.2072453.

[MPK10] B. Meixner, F. Pein, and H. Kosch. “Flexible optimization of text recognition algorithms”. In: *2010 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. IEEE, 2010, pp. 156–161. DOI: 10.1109/SOCPAR.2010.5685975.

[Mei+10a] B. Meixner, B. Siegel, G. Hölbling, F. Lehner, and H. Kosch. “SIVA Suite - authoring system and player for interactive non-linear videos”. In: *Proceedings of the International Conference on Multimedia. MM ’10*. New York, New York, USA: ACM, 2010, pp. 1563–1566. DOI: 10.1145/1873951.1874287.

[Mei+10b] B. Meixner, J.-O. Janda, B. Siegel, G. Hölbling, H. Kosch, and F. Lehner. “XML-Schema für interaktive Videos”. In: *Workshop Audiovisuelle Medien WAM 2010*. Ed. by M. Eibl. Website (accessed April 26, 2014). Chemnitz, 2010. URL: http://dl.mensch-und-computer.de/handle/123456789/3211.

[Mei+11a] B. Meixner, K. Kandlbinder, B. Siegel, H. Kosch, and F. Lehner. “Player zur Wiedergabe von erweiterten Videos - Bewertung durch Guidelines”. In: *Mensch & Computer 2011: überMEDIEN|ÜBERmorgen*. Ed. by M. Eibl. Website (accessed April 26, 2014). München: Oldenbourg Verlag, 2011, pp. 319–322. URL: http://dl.mensch-und-computer.de/handle/123456789/1527.

[Mei+11b] B. Meixner, B. Siegel, P. Schultes, G. Hölbling, F. Lehner, and H. Kosch. *SIVA Suite - Simple Interactive Video Authoring Suite. Produkt- und Projektdokumentation*. Technical Report. Passau: University of Passau, 2011.

[Mei+12a] B. Meixner, B. Siegel, C. Grill, A. Lang, F. Lehner, and H. Kosch. *SIVA Suite - Usability Report, Diskussionsbeitrag Nr. W-37-12, Schriftenreihe Wirtschaftsinformatik, Passauer Diskussionspapiere, ISSN 1613-8252*. Passau: University of Passau, 2012.

[Mei+12b] B. Meixner, K. Matusik, C. Grill, and H. Kosch. “Towards an easy to use authoring tool for interactive non-linear video”. In: *Multimedia Tools and Applications* (2012), pp. 1–26. DOI: 10.1007/s11042-012-1218-6.

[Mei+12c] B. Meixner, K. Kandlbinder, B. Siegel, F. Lehner, H. Kosch, and A. Kohl. “What users expect from players for interactive (non-linear) videos”. In: *Mensch & Computer 2012: interaktiv informiert - allgegenwärtig und allumfassend!?* Ed. by H. Reiterer and O. Deussen. Website (accessed April 26, 2014). München: Oldenbourg Verlag, 2012, pp. 337–340. URL: http://dl.mensch-und-computer.de/handle/123456789/2906.

[Mei+13] B. Meixner, B. Siegel, P. Schultes, F. Lehner, and H. Kosch. “An HTML5 player for interactive non-linear video with time-based collaborative annotations”. In: *Proceedings of International Conference on Advances in Mobile Computing & Multimedia. MoMM ’13*. Ed. by R. Mayrhofer, L. Chen, M. Steinbauer, I. Khalil, and G. Kotsis. New York, NY, USA: ACM, 2013, pp. 490–498. DOI: 10.1145/2536853.2536868.

[Mem10] I. Memruk. *Android architecture: message-based MVC*. Website (accessed April 26, 2014). 2010. URL: http://mindtherobot.com/blog/675/android-architecture-message-based-mvc/.

[MRLD08] J. Mikác, C. Roisin, and B. Le Duc. “An export architecture for a multimedia authoring environment”. In: *Proceedings of the 8th ACM Symposium on Document Engineering. DocEng ’08*. New York, NY, USA: ACM, 2008, pp. 28–31. DOI: 10.1145/1410140.1410147.

[Mil92] F. W. Miller. “The performance of a mixed priority real-time scheduling algorithm”. In: *SIGOPS Operating Systems Review* 26.4 (1992), pp. 5–13. DOI: 10.1145/142854.142857.

[Mil+11] G. Miller, S. Fels, M. Ilich, M. M. Finke, T. Bauer, K. Wong, and S. Mueller. “An end-to-end framework for multi-view video content: creating multiple-perspective hypervideo to view on mobile platforms”. In: *Entertainment Computing - ICEC 2011*. Ed. by J. C. Anacleto, S. Fels, N. Graham, B. Kapralos, M. Saif El-Nasr, and K. Stanley. Vol. 6972. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 337–342. DOI: 10.1007/978-3-642-24500-8_37.

[Mil89] R. Milner. *Communication and concurrency*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[Mir+11] S. Mirri, L. A. Muratori, M. Roccetti, and P. Salomoni. “The Directors’ cut: a solution to collaborative multimedia management”. In: *Multimedia Tools and Applications* 53.1 (2011), pp. 319–344. DOI: 10.1007/s11042-010-0533-z.

[Mon09] J. Monaco. *How to read a film: movies, media, and beyond*. 4, illustr. Oxford University Press, 2009.

[MB02] A. Mostefaoui and L. Brunie. “Optimizing server I/O for multimedia presentations”. In: *2002 IEEE International Conference on Multimedia and Expo, 2002. ICME ’02. Proceedings.* Vol. 1. IEEE, 2002, pp. 21–24. DOI: 10.1109/ICME.2002.1035708.

[Mou00] C. Mourlas. “Predictability and resource management in distributed multimedia presentations”. In: *Parallel and Distributed Processing*. Ed. by J. Rolim. Vol. 1800. Lecture Notes in Computer Science. London, UK, UK: Springer Berlin Heidelberg, 2000, pp. 750–756. DOI: 10.1007/3-540-45591-4_103.

[Mou02] C. Mourlas. “Specification and verification of quality requirements in distributed multimedia”. In: *22nd International Conference on Distributed Computing Systems Workshops, 2002. Proceedings*. IEEE, 2002, pp. 323–328. DOI: 10.1109/ICDCSW.2002.1030789.

[Moz13a] Mozilla Corporation. *Popcorn Maker*. Website (accessed April 26, 2014). 2013. URL: https://popcorn.webmaker.org.

[Moz13b] Mozilla Developer Network and individual contributors. *JavaScript*. Website (accessed April 26, 2014). 2013. URL: https://developer.mozilla.org/en-US/docs/JavaScript.

[MSRS02] D. C. Muchaluat-Saade, R. F. Rodrigues, and L. F. G. Soares. “XConnector: extending XLink to provide multimedia synchronization”. In: *Proceedings of the 2002 ACM Symposium on Document Engineering. DocEng ’02*. New York, NY, USA: ACM, 2002, pp. 49–56. DOI: 10.1145/585058.585069.

[MD07] S. Mujacic and M. Debevc. “A formal approach to hypervideo design”. In: *14th International Workshop on Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*. IEEE, 2007, pp. 189 –192. DOI: 10.1109/IWSSIP.2007.4381185.

[Muj+12] S. Mujacic, M. Debevc, P. Kosec, M. Bloice, and A. Holzinger. “Modeling, design, development and evaluation of a hypervideo presentation for digital systems teaching and learning”. In: *Multimedia Tools and Applications* 58.2 (2012), pp. 435–452. DOI: 10.1007/s11042-010-0665-1.

[MAJ98] C. D. Murta, V. A. F. Almeida, and W. M. Jr. “Analyzing performance of partitioned caches for the WWW”. In: *3rd International WWW Caching Workshop and TERENA TF-Cache Meeting*. Website (accessed April 26, 2014). 1998. URL: http://iwcw.ircache.net/1998/24/.

[Mur+06] U. Murthy, K. Ahuja, S. Murthy, and E. A. Fox. “SIMPEL: a superimposed multimedia presentation editor and player”. In: *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, 2006. JCDL ’06*. IEEE, 2006, pp. 377–377. DOI: 10.1145/1141753.1141873.

[NT11] A. T. Naman and D. Taubman. “JPEG2000-based scalable interactive video (JSIV)”. In: *IEEE Transactions on Image Processing* 20.5 (2011), pp. 1435–1449. DOI: 10.1109/TIP.2010.2093905.

[Nap+01] M. R. Naphade, C.-Y. Lin, J. R. Smith, B. L. Tseng, and S. Basu. “Learning to annotate video databases”. In: *Proceedings of SPIE*. Vol. 4676. SPIE, 2001, pp. 264–175. DOI: 10.1117/12.451096.

[NC10] L. A. R. Neng and T. Chambel. “Get around 360&Deg; hypervideo”. In: *Proceedings of the 14th International Academic MindTrek Conference on Envisioning Future Media Environments. MindTrek ’10*. New York, New York, USA: ACM, 2010, pp. 119–122. DOI: 10.1145/1930488.1930512.

[NKN91] S. R. Newcomb, N. A. Kipp, and V. T. Newcomb. “The HyTime: hypermedia/time-based document structuring language”. In: *Communications of the ACM* 34.11 (1991), pp. 67–83. DOI: 10.1145/125490.125495.

[NLN98] N. Niclausse, Z. Liu, and P. Nain. “A new efficient caching policy for the World Wide Web”. In: *Proc. Workshop on Internet Server Performance (WISP ’98)*. Website (accessed April 26, 2014). 1998. URL: http://www-sop.inria.fr/members/Philippe.Nain/PAPERS/WEB/proxy_cache_final.pdf.

[NKL09] Y. Nimmagadda, K. Kumar, and Y.-H. Lu. “Preference-based adaptation of multimedia presentations for different display sizes”. In: *2009 IEEE International Conference on Multimedia and Expo. ICME 2009*. IEEE, 2009, pp. 978–981. DOI: 10.1109/ICME.2009.5202660.

[NBB13] L. Nixon, M. Bauer, and C. Bara. “Connected media experiences: interactive video using Linked Data on the Web”. In: *Proceedings of the 22nd International Conference on World Wide Web Companion. WWW ’13 Companion*. Website (accessed April 26, 2014). Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 309–

312. URL: http://www2013.wwwconference.org/companion/p309.pdf.

[OMN13] OMNeT++ Community. *OMNeT++*. Website (accessed April 26, 2014). 2013. URL: http://www.omnetpp.org/.

[OL13] D. Ockeloen and P. van Leeuwen. *Deliverable 3.6: Interface and presentation engine version 2*. Website (accessed December 27, 2013). 2013. URL: http://www.linkedtv.eu/wp/wp-content/uploads/2013/12/LinkedTV_D3.6.pdf.

[Oeh+13] P. Oehme, M. Krug, F. Wiedemann, and M. Gaedke. "The Chrooma+ approach to enrich video content using HTML5". In: *Proceedings of the 22nd International Conference on World Wide Web Companion. WWW '13 Companion*. Website (accessed April 26, 2014). Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 479–480. URL: http://www2013.wwwconference.org/companion/p479.pdf.

[Oga+92] R. Ogawa, E. Tanaka, D. Taguchi, and K. Harada. "Design strategies for scenario-based hypermedia: description of its structure, dynamics, and style". In: *Proceedings of the ACM Conference on Hypertext. ECHT '92*. New York, NY, USA: ACM, 1992, pp. 71–80. DOI: 10.1145/168466.168494.

[Ora] Oracle. *Learn about Java technology*. Website (accessed April 26, 2014). URL: https://www.java.com/en/about/.

[Ove10] Overlay TV Inc. *Overlay.TV*. Website (accessed April 26, 2014). 2010. URL: http://www.overlay.tv/.

[OHK96] G. Ozsoyoglu, V. Hakkoymaz, and J. D. Kraft. "Automating the assembly of presentations from multimedia databases". In: *Proceedings of the 12th International Conference on Data Engineering, 1996*. IEEE, 1996, pp. 593–601. DOI: 10.1109/ICDE.1996.492210.

[Pal+11] C. E. Palau, J. Mares, B. Molina, and M. Esteve. "Wireless CDN video streaming architecture for IPTV". In: *Multimedia Tools and Applications* 53.3 (2011), pp. 591–613. DOI: 10.1007/s11042-010-0516-0.

[PP10] J. M. Paluska and H. Pham. "Interactive streaming of structured data". In: *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2010, pp. 11–19. DOI: 10.1109/PERCOM.2010.5466996.

[Pan+04] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. J. Wu, and S. Wu. "Proactive frame-skipping decision scheme for variable frame rate video coding". In: *2004 IEEE International Conference on Multimedia and Expo, 2004. ICME '04.* Vol. 3. IEEE, 2004, pp. 1903–1906. DOI: 10.1109/ICME.2004.1394631.

[Pan+12] M. Panka, M. Chlebiej, K. Benedyczak, and P. Bala. “Distributed collaborative visualization on mobile devices using interactive video streaming techniques”. In: *Parallel Processing and Applied Mathematics*. Ed. by R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski. Vol. 7204. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 191–200. DOI: 10.1007/978-3-642-31500-8_20.

[Par+11] A. ParandehGheibi, M. Medard, A. Ozdaglar, and S. Shakkottai. “Avoiding interruptions - A QoE reliability function for streaming media applications”. In: *IEEE Journal on Selected Areas in Communications* 29.5 (2011), pp. 1064–1074. DOI: 10.1109/JSAC.2011.110516.

[PG93] A. K. Parekh and R. G. Gallager. “A generalized processor sharing approach to flow control in integrated services networks: the single-node case”. In: *IEEE/ACM Transactions on Networking* 1.3 (1993), pp. 344–357. DOI: 10.1109/90.234856.

[Par+07] Y. Park, Y. Lee, H. Kim, and K. Kim. “Segment based caching replacement algorithm in streaming media transcoding proxy”. In: *Managing Next Generation Networks and Services*. Ed. by S. Ata and C. Hong. Vol. 4773. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 612–615. DOI: 10.1007/978-3-540-75476-3_78.

[Par83] E. Parsloe. “Interactive video”. In: *Journal of European Industrial Training* 7.3 (1983), pp. 28–32. DOI: 10.1108/eb002148.

[PJT06] N. Pattanasri, A. Jatowt, and K. Tanaka. “Enhancing comprehension of events in video through explanation-on-demand hypervideo”. In: *Advances in Multimedia Modeling*. Ed. by T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, and L.-T. Chia. Vol. 4351. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 535–544. DOI: 10.1007/978-3-540-69423-6_52.

[Pee10] H. B. Peek. “The emergence of the compact disc”. In: *IEEE Communications Magazine* 48.1 (2010), pp. 10–17. DOI: 10.1109/MCOM.2010.5394021.

[Pei10] F. Pein. “Evaluation, Implementierung und Adaption von Methoden der Texterkennung”. Thesis. Passau, Germany: Universität Passau, 2010.

[Pet+06] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, Y. Kompatsiaris, and S. Staab. “M-OntoMat-Annotizer: image annotation linking ontologies and multimedia low-level features”. In: *Knowledge-Based Intelligent Information and Engineering Systems*. Ed. by B. Gabrys, R. J. Howlett, and L. C. Jain. Vol. 4253. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 633–640. DOI: 10.1007/11893011_80.

[PV06] K. Pihkala and P. Vuorimaa. “Nine methods to extend SMIL for multimedia applications”. In: *Multimedia Tools and Applications* 28.1 (2006), pp. 51–67. DOI: 10.1007/s11042-006-5120-y.

[PB03] S. Podlipnig and L. Böszörmenyi. “A survey of Web cache replacement strategies”. In: *ACM Computing Surveys* 35.4 (2003), pp. 374–398. DOI: 10.1145/954339.954341.

[PS09] C. N. Potts and V. A. Strusevich. “Fifty years of scheduling: a survey of milestones”. In: *Journal of the Operational Research Society* 60.1 (2009), pp. 41–68. DOI: 10.1057/jors.2009.2.

[Pra00] B. Prabhakaran. “Adaptive multimedia presentation strategies”. In: *Multimedia Tools and Applications* 12.2-3 (2000), pp. 281–298. DOI: 10.1023/A:1009627926302.

[PBD02] S. L. Presti, D. Bert, and A. Duda. “TAO: temporal algebraic operators for modeling multimedia presentations”. In: *Journal of Network and Computer Applications* 25.4 (2002), pp. 319–342. DOI: 10.1006/jnca.2002.0135.

[Pro12] Project for Excellence in Journalism. *YouTube & news: video length*. Website (accessed July 22, 2013). 2012. URL: http://www.journalism.org/analysis_report/video_length\#_ftnref3.

[Pro02] W. Provost. *UML for W3C XML schema design*. Website (accessed April 26, 2014). 2002. URL: http://www.xml.com/pub/a/2002/08/07/wxs_uml.html.

[QD06] Y. Qi and M. Dai. “The Effect of Frame Freezing and Frame Skipping on Video Quality”. In: *International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP ’06.* 2006, pp. 423–426. DOI: 10.1109/IIH-MSP.2006.265032.

[Qui10] Quick TV Limited. *Make your video dynamic & interactive*. Website (accessed March 26, 2013). 2010. URL: http://www.quick.tv/.

[RDF04] RDF Working Group. *Resource description framework (RDF)*. Standard-/Website (accessed April 26, 2014). 2004. URL: http://www.w3.org/RDF/.

[RSA10] C. Räck, R. Seeliger, and S. Arbanowski. “Featured media: nonlinear, clickable multimedia content for an interactive video experience”. In: *2010 2nd International Conference on Advances in Multimedia (MMEDIA)*. IEEE, 2010, pp. 39–43. DOI: 10.1109/MMEDIA.2010.18.

[Ram+99] A. Ram, R. Catrambone, M. J. Guzdial, C. M. Kehoe, D. S. McCrickard, and J. T. Stasko. “PML: adding flexibility to multimedia presentations”. In: *IEEE MultiMedia* 6.2 (1999), pp. 40–52. DOI: 10.1109/93.771372.

[Rea12] RealNetworks, Inc. *RealPlayer*. Website (accessed April 26, 2014). 2012. URL: http://eu.real.com/features/basic/playback.html.

[RGT13] L. J. Redondo-Garcia and R. Troncy. “Television meets the Web: a multimedia hypervideo experience”. In: *Proceedings of the Doctoral Consortium co-located with 12th International Semantic Web Conference (ISWC 2013)*. Ed. by L. Aroyo and N. Noy. Website (accessed April 26, 2014). 2013, pp. 48–55. URL: http://ceur-ws.org/Vol-1045/paper-07.pdf.

[Ree10] ReelSEO. *Interactive and clickable video drives e-commerce conversions*. Website (accessed April 26, 2014). 2010. URL: http://www.reelseo.com/interactive-video-conversions.

[Ree11] ReelSEO. *How to overcome the pitfalls of interactive, clickable video*. Website (accessed April 26, 2014). 2011. URL: http://www.reelseo.com/pitfalls-clickable-video.

[Riv13] Riverbed Technology. *Network simulation (OPNET Modeler Suite)*. Website (accessed April 26, 2014). 2013. URL: http://www.riverbed.com/products-solutions/products/network-planning-simulation/Network-Simulation.html.

[RV00] L. Rizzo and L. Vicisano. "Replacement policies for a proxy cache". In: *IEEE/ACM Transactions on Networking* 8.2 (2000), pp. 158–170. DOI: 10.1109/90.842139.

[Rob07] R. Robberecht. "Interactive nonlinear learning environments". In: *The Electronic Journal of e-Learning* 5.1 (2007), pp. 59–68.

[Ros+93] G. van Rossum, J. Jansen, K. S. Mullender, and D. C. A. Bulterman. "CMIFed: a presentation environment for portable hypermedia documents". In: *Proceedings of the 1st ACM International Conference on Multimedia. MULTIMEDIA '93*. New York, NY, USA: ACM, 1993, pp. 183–188. DOI: 10.1145/166266.166287.

[Rou+99] F. Rousseau, J. A. García-Marcías, J. Valdeni de Lima, and A. Duda. "User adaptable multimedia presentations for the World Wide Web". In: *Proceedings of the 8th International Conference on World Wide Web. WWW '99*. Website (accessed April 26, 2014). New York, NY, USA: Elsevier, 1999, pp. 1273–1290. URL: http://www.ambuehler.ethz.ch/cdstore/www8/data/2162/pdf/pd1.pdf.

[Rut+98] L. Rutledge, L. Hardman, T. van Ossenbruggen, and D. C. A. Bulterman. "Implementing adaptability in the standard reference model for intelligent multimedia presentation systems". In: *MultiMedia Modeling, 1998. MMM '98. Proceedings*. IEEE, 1998, pp. 12–20. DOI: 10.1109/MULMM.1998.722969.

[SW10] H. Sack and J. Waitelonis. "Exploratory semantic video search with yovisto". In: *2010 IEEE 4th International Conference on Semantic Computing (ICSC)*. IEEE, 2010, pp. 446–447. DOI: 10.1109/ICSC.2010.98.

[SAP11] M. Sadallah, O. Aubert, and Y. Prié. "Component-based hypervideo model: high-level operational specification of hypervideos". In: *Proceedings of the 11th ACM Symposium on Document Engineering. DocEng '11*. New York, NY, USA: ACM, 2011, pp. 53–56. DOI: 10.1145/2034691.2034701.

[SD94] P. Salant and D. Dillman. *How to conduct your own survey*. Business Reference: Wiley, 1994.

[SSC00] P. N. M. Sampaio, C. A. S. Santos, and J. P. Courtias. "About the semantic verification of SMIL documents". In: *2000 IEEE International Conference on Multimedia and Expo, 2000. ICME 2000*. Vol. 3. IEEE, 2000, pp. 1675–1678. DOI: 10.1109/ICME.2000.871093.

[Sap02] M. L. Sapino. “Multimedia presentations databases (extended abstract)”. In: *SOFSEM 2002: Theory and Practice of Informatics*. Ed. by W. I. Grosky and F. Plasil. Vol. 2540. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 132–134. DOI: 10.1007/3-540-36137-5_7.

[SAAH10] N. J. Sarhan, M. A. Alsmirat, and M. Al-Hadrusi. “Waiting-time prediction in scalable on-demand video streaming”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 6.2 (2010), 11:1–11:25. DOI: 10.1145/1671962.1671967.

[SBS96] N. Sawhney, D. Balcom, and I. Smith. “HyperCafe: narrative and aesthetic properties of hypervideo”. In: *Proceedings of the 7th ACM Conference on Hypertext. HYPERTEXT ’96*. New York, NY, USA: ACM, 1996, pp. 1–10. DOI: 10.1145/234828.234829.

[SBS97] N. Sawhney, D. Balcom, and I. Smith. “Authoring and navigating video in space and time”. In: *IEEE Multimedia* 4.4 (1997), pp. 30–39. DOI: 10.1109/93.641877.

[SB05] A. Scherp and S. Boll. “Paving the last mile for multi-channel multimedia presentation generation”. In: *Proceedings of the 11th International Multimedia Modelling Conference, 2005. MMM 2005*. IEEE, 2005, pp. 190–197. DOI: 10.1109/MMMC.2005.58.

[SSV97] P. Scheuermann, J. Shim, and R. Vingralek. “A case for delay-conscious caching of Web documents”. In: *Computer Networks and ISDN Systems* 29.8-13 (1997), pp. 997–1005. DOI: 10.1016/S0169-7552(97)00032-9.

[SW94] G. Schloss and M. Wynblatt. “Building temporal structures in a layered multimedia data model”. In: *Proceedings of the 2nd ACM International Conference on Multimedia. MULTIMEDIA ’94*. New York, New York, USA: ACM, 1994, pp. 271–278. DOI: 10.1145/192593.192674.

[SYS98] P. Schmitz, J. Yu, and P. Santangeli. *Timed interactive multimedia extensions for HTML (HTML+TIME): extending SMIL into the Web browser*. Website (accessed April 26, 2014). 1998. URL: http://www.w3.org/TR/NOTE-HTMLplusTIME.

[SBH03] O. Schneider, N. Braun, and G. Habinger. “Storylining suspense: an authoring environment for structuring non-linear interactive narratives”. In: *Journal of WSCG* 11.1 (2003). Website (accessed April 26, 2014), pp. 411–417. URL: http://wscg.zcu.cz/wscg2003/Papers_2003/I53.pdf.

[SKD96] J. Schnepf, J. A. Konstan, and D. H.-C. Du. “Doing FLIPS: flexible interactive presentation synchronization”. In: *IEEE Journal on Selected Areas in Communications* 14.1 (1996), pp. 114–125. DOI: 10.1109/49.481698.

[See10] R. Seeliger. “Non-linear video”. In: *CONTENT 2010, The Second International Conference on Creative Content Technologies*. Website (accessed April 26, 2014). Lisbon, Portugal, 2010, pp. 34–38. URL: http://www.thinkmind.org/index.php?view=article&articleid=content_2010_2_10_60001.

[Sei11] N. Seidel. “Enable Wikis for seamless hypervideo integration”. In: *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics. ECCE ’11*. New York, NY, USA: ACM, 2011, pp. 251–252. DOI: 10.1145/2074712.2074765.

[Sei+09] F. H. Seitner, M. Bleyer, M. Gelautz, and R. M. Beuschel. “Development of a high-level simulation approach and its application to multicore video decoding”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 19.11 (2009), pp. 1667–1679. DOI: 10.1109/TCSVT.2009.2031523.

[Sei+11] F. H. Seitner, M. Bleyer, M. Gelautz, and R. M. Beuschel. “Evaluation of data-parallel H.264 decoding approaches for strongly resource-restricted architectures”. In: *Multimedia Tools and Applications* 53.2 (2011), pp. 431–457. DOI: 10.1007/s11042-010-0501-7.

[Sen09] S. H. Sengamedu. “Bookmarking in videos”. Pat. 20090067806. 2009. URL: http://www.freepatentsonline.com/y2009/0067806.html.

[SGK12] D. Shabtay, N. Gaspar, and M. Kaspi. “A survey on offline scheduling with rejection”. In: *Journal of Scheduling* 16.1 (2012), pp. 3–28. DOI: 10.1007/s10951-012-0303-z.

[Sha+07] R. Sharman, S. S. Ramanna, R. Ramesh, and R. Gopal. “Cache architecture for on-demand streaming on the Web”. In: *ACM Transactions on the Web* 1.3 (2007), 13.1–13:35. DOI: 10.1145/1281480.1281483.

[She07] D. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Fourth Edition. Chapman & Hall/CRC, 2007.

[Shi+09] L. Shi, J. Pang, L. Yang, T. Zhang, and D. Wang. “Fair-priority-expression-based burst scheduling to enhance performance and fairness of shared dram systems”. In: *Canadian Conference on Electrical and Computer Engineering, 2009. CCECE ’09*. IEEE, 2009, pp. 190–194. DOI: 10.1109/CCECE.2009.5090118.

[SD97] T. K. Shih and R. E. Davis. “IMMPS: a multimedia presentation design system”. In: *IEEE Multimedia* 4.2 (1997), pp. 67–78. DOI: 10.1109/93.591175.

[Shi+98a] T. K. Shih, Y. H. Wang, C. H. Kuo, L. Y. Deng, D. R. Jiang, W. C. Pai, and C. C. Wang. “A spatial/temporal relation computing technology for multimedia presentation designs”. In: *Proceedings of the 31st Hawaii International Conference on System Sciences, 1998*. Vol. 2. IEEE, 1998, pp. 27–36. DOI: 10.1109/HICSS.1998.651680.

[Shi97] T. K. Shih. “Case studies in intelligent multimedia presentation design systems in terms of the standard reference model: the IMMPS project, the PreGen system, and the StrMP system”. In: *Computer Standards & Interfaces* 18.6-7 (1997), pp. 605–612. DOI: 10.1016/S0920-5489(97)00024-X.

[Shi98] T. K. Shih. “Participator dependent multimedia presentation”. In: *Information Sciences* 107.1-4 (1998), pp. 85–105. DOI: 10.1016/S0020-0255(97)10043-3.

[Shi+98b] T. K. Shih, Y. H. Wang, C. H. Kuo, D. R. Jiang, J. C. Hung, W. C. Pai, and C. C. Wang. “A software engineering approach to multimedia presentation designs”. In: *Proceedings of the 31st Hawaii International Conference on System Sciences, 1998*. Vol. 2. IEEE, 1998, pp. 37–46. DOI: 10.1109/HICSS.1998.651681.

[Shi+99] T. K. Shih, H.-C. Keh, Y.-H. Wang, and Y.-F. Kuo. “Temporal properties underlying multimedia presentations with Z Notations”. In: *Journal of Information Science and Engineering* 15.1 (1999). Website (accessed April 26, 2014), pp. 107–129. URL: http://www.iis.sinica.edu.tw/page/jise/1999/199901_08.pdf.

[Shi13] D.-H. Shin. “Defining sociability and social presence in Social TV”. In: *Computers in Human Behavior* 29.3 (2013), pp. 939–947. DOI: 10.1016/j.chb.2012.07.006.

[SGW03a] F. Shipman, A. Girgensohn, and L. Wilcox. “Combining spatial and navigational structure in the hyper-hitchcock hypervideo editor”. In: *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia. HYPERTEXT ’03*. New York, NY, USA: ACM, 2003, pp. 124–125. DOI: 10.1145/900074.900078.

[SGW03b] F. Shipman, A. Girgensohn, and L. Wilcox. “Generation of interactive multi-level video summaries”. In: *Proceedings of the 11th ACM International Conference on Multimedia. MULTIMEDIA ’03*. New York, NY, USA: ACM, 2003, pp. 392–401. DOI: 10.1145/957092.957096.

[SGW03c] F. Shipman, A. Girgensohn, and L. Wilcox. “Hyper-Hitchcock: towards the easy authoring of interactive video”. In: *Human-Computer Interaction. INTERACT ’03: IFIP TC13*. Ed. by M. Rauterberg, M. Menozzi, and J. Wesson. Website (accessed April 26, 2014). IOS Press, 2003. URL: http://www.fxpal.com/publications/FXPAL-PR-03-219.pdf.

[SGW05] F. Shipman, A. Girgensohn, and L. Wilcox. “Hypervideo expression: experiences with Hyper-Hitchcock”. In: *Proceedings of the 16th ACM Conference on Hypertext and Hypermedia. HYPERTEXT ’05*. New York, NY, USA: ACM, 2005, pp. 217–226. DOI: 10.1145/1083356.1083401.

[SGW08] F. Shipman, A. Girgensohn, and L. Wilcox. “Authoring, viewing, and generating hypervideo: an overview of Hyper-Hitchcock”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 5.2 (2008), 15:1–15:19. DOI: 10.1145/1413862.1413868.

[Shi85] Y. Shiraishi. “History of home videotape recorder development”. In: *SMPTE Motion Imaging Journal* 94.12 (1985), pp. 1257–1263. DOI: 10.5594/J03317.

[SV95] M. Shreedhar and G. Varghese. “Efficient fair queueing using deficit round robin”. In: *ACM SIGCOMM Computer Communication Review* 25.4 (1995), pp. 231–242. DOI: 10.1145/217391.217453.

[Sht08] Y. Shterev. “Performance of multimedia presentation with branches by synchronized multimedia integration language”. In: *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing. CompSysTech ’08*. New York, NY, USA: ACM, 2008, 24:IIIA.4–24:1. DOI: 10.1145/1500879.1500907.

[Shu+93] J.-S. Shue, C.-H. Hsieh, H.-S. Tsai, and C.-C. Wang. "Variable-rate video codec using frame adaptive finite-state vector quantization". In: *1993 IEEE International Symposium on Circuits and Systems, 1993. ISCAS '93*. Vol. 1. IEEE, 1993, pp. 28–31. DOI: 10.1109/ISCAS.1993.393647.

[SC56] S. Siegel and N. J. Castellan. *Nonparametric statistics for the behavioral sciences*. Reprint 1988. McGraw-Hill series in psychology. McGraw-Hill Inc., 1956.

[Sil+04] H. V. O. Silva, R. F. Rodrigues, L. F. G. Soares, and D. C. Muchaluat Saade. "NCL 2.0: integrating new concepts to XML modular languages". In: *Proceedings of the 2004 ACM Symposium on Document Engineering. DocEng '04*. New York, NY, USA: ACM, 2004, pp. 188–197. DOI: 10.1145/1030397.1030433.

[Sin+11] V. Singh, C. Latulipe, E. Carroll, and D. Lottridge. "The choreographer's notebook: a video annotation system for dancers and choreographers". In: *Proceedings of the 8th ACM Conference on Creativity and Cognition. C&C'11*. New York, NY, USA: ACM, 2011, pp. 197–206. DOI: 10.1145/2069618.2069653.

[Smi87] E. E. Smith. "Interactive video: an examination of use and effectiveness". English. In: *Journal of Instructional Development* 10.2 (1987), pp. 2–10. DOI: 10.1007/BF02905785.

[Sny03] C. Snyder. *Paper prototyping: the fast and easy way to design and refine user interfaces*. The Morgan Kaufmann Series in Interactive Technologies: Morgan Kaufmann, 2003.

[SIJT10] C. So-In, R. Jain, and A.-K. A. Tamimi. "Deficit round robin with fragmentation scheduling to achieve generalized weighted fairness for resource allocation in IEEE 802.16e mobile WiMAX networks". In: *Future Internet* 2.4 (2010), pp. 446–468. DOI: 10.3390/fi2040446.

[SRMS00] L. F. G. Soares, R. F. Rodrigues, and D. C. Muchaluat Saade. "Modeling, authoring and formatting hypermedia documents in the HyperProp system". In: *Multimedia Systems* 8.2 (2000), pp. 118–134. DOI: 10.1007/s005300050155.

[Soa+10] L. F. G. Soares, R. F. Rodrigues, R. Cerqueira, and S. D. J. Barbosa. "Variable and state handling in NCL". In: *Multimedia Tools and Applications* 50.3 (2010), pp. 465–489. DOI: 10.1007/s11042-010-0478-2.

[SR05] L. F. G. Soares and R. F. Rodrigues. *Nested Context Model 3.0. Part 1 - NCM Core*. 18. Technical Report. Website (accessed April 26, 2014). Rio de Janeiro: Laboratorio TeleMidia DI - PUC-Rio, 2005. URL: http://www.telemidia.puc-rio.br/sites/telemidia.puc-rio.br/files/Part1-NCM3.0.pdf.

[SK97] J. M. Sohn and G. Y. Kim. "Earliest-deadline-first scheduling on non-preemptive real-time threads for a continuous media server". In: *High-Performance Computing and Networking*. Ed. by B. Hertzberger and P. Sloot. Vol. 1225. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1997, pp. 950–956. DOI: 10.1007/BFb0031666.

[SBS94] M. W. van Someren, Y. F. Barnard, and J. A. C. Sandberg. *The Think Aloud Method: a practical guide to modelling cognitive processes*. illustrated. Knowledge-Based Systems. London: Academic Press, 1994.

[SLC08] B. Song, Y.-H. Lin, and R. L. Cruz. "Weighted max-min fair beamforming, power control, and scheduling for a MISO downlink". In: *IEEE Transactions on Wireless Communications* 7.2 (2008), pp. 464–469. DOI: 10.1109/TWC.2007.060675.

[Spa+06] M. Spaniol, R. Klamma, N. Sharda, and M. Jarke. "Web-based learning with non-linear multimedia stories". In: *Advances in Web Based Learning - ICWL 2006*. Ed. by W. Liu, Q. Li, and R. W.H. Lau. Vol. 4181. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 249–263. DOI: 10.1007/11925293_23.

[Spa+12] I. A. E. Spanjers, T. van Gog, P. Wouters, and J. J. G. van Merriënboer. "Explaining the segmentation effect in learning from animations: The role of pausing and temporal cueing". In: *Computers & Education* 59.2 (2012), pp. 274–280. DOI: 10.1016/j.compedu.2011.12.024.

[Spe11] A. Spendel. "Konzeption, Implementierung und Evaluation eines HTML5-Players für interaktive Videos mit Loggingfunktionalität". Thesis. Passau, Germany: Universität Passau, 2011.

[Spi+12] R. Spicer, Y.-R. Lin, A. Kelliher, and H. Sundaram. "NextSlidePlease: authoring and delivering agile multimedia presentations". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 8.4 (2012), 53:1–53:20. DOI: 10.1145/2379790.2379795.

[SZF05] E. Stahl, C. Zahn, and M. Finke. "How can we use hypervideo design projects to construct knowledge in university courses?" In: *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning 2005: the Next 10 Years!. CSCL '05*. Morristown, NJ, USA: International Society of the Learning Sciences, 2005, pp. 641–646. DOI: 10.3115/1149293.1149377.

[SE96] M. K. Stenzler and R. R. Eckert. "Interactive video". In: *ACM SIGCHI Bulletin* 28.2 (1996), pp. 76–81. DOI: 10.1145/226650.226676.

[SV98] D. Stiliadis and A. Varma. "Efficient fair queueing algorithms for packet-switched networks". In: *IEEE/ACM Transactions on Networking* 6.2 (1998), pp. 175–185. DOI: 10.1109/90.664266.

[SW12] H.-M. Sun and M.-W. Weng. "Rate-smoothed schedule with tolerable data dropping for video coding stream". In: *Multimedia Tools and Applications* 57.3 (2012), pp. 587–604. DOI: 10.1007/s11042-010-0659-z.

[SL05] M. Y. Sung and D. H. Lee. "A Java-based collaborative authoring system for multimedia presentation". In: *Advances in Multimedia Information Processing - PCM 2004*. Ed. by K. Aizawa, Y. Nakamura, and S. Satoh. Vol. 3332. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 96–103. DOI: 10.1007/978-3-540-30542-2_13.

[Sys10] Sysomos Inc. *Inside YouTube videos: exploring YouTube videos and their use in blogosphere - Michael Jackson and health care dominate.* Website (accessed April 26, 2014). 2010. URL: http://www.sysomos.com/reports/youtube/.

[TG05] R. Tan and S.-U. Guan. "A dynamic Petri net model for iterative and interactive distributed multimedia presentation". In: *IEEE Transactions on Multimedia* 7.5 (2005), pp. 869–879. DOI: 10.1109/TMM.2005.854377.

[Tat98] I. Tatarinov. *An efficient LFU-like policy for web caches*. Technical Report. Website (accessed April 26, 2014). North Dakota State University, Wahpeton, ND: Computer Science Department, 1998. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.4672&rep=rep1&type=pdf.

[Tay99] J. Taylor. "DVD-Video: Multimedia for the masses". In: *IEEE Multimedia* 6 (1999), pp. 86–92. DOI: 10.1109/93.790615.

[Tel11] TeleMidia Lab - PUC-Rio. *NCL - Nested Context Language*. Standard/Website (accessed April 26, 2014). 2011. URL: http://www.ncl.org.br/en/inicio.

[Tel88] E. R. Tello. *Mastering Ai tools and techniques: for the IBM Pc, At, Pc/2, and compatibles*. Howard W. Sams & Company, 1988.

[Ter+00] Y. Terashima, K. Yasumoto, T. Higashinoz, K. Abe, T. Matsuura, and K. Taniguchi. "Extension of SMIL with QoS control and its implementation". In: *2000 IEEE International Conference on Multimedia and Expo, 2000. ICME 2000*. Vol. 3. IEEE, 2000, pp. 1683–1686. DOI: 10.1109/ICME.2000.871095.

[The] The University of Southern California. *The network simulator - ns-2*. Website (accessed April 26, 2014). URL: http://www.isi.edu/nsnam/ns/.

[Tie+10] C. A. B. Tiellet, A. G. Pereira, E. B. Reategui, J. V. Lima, and T. Chambel. "Design and evaluation of a hypervideo environment to support veterinary surgery learning". In: *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia. HT '10*. New York, New York, USA: ACM, 2010, pp. 213–222. DOI: 10.1145/1810617.1810656.

[Ton+12] K. Tonndorf, T. Knieper, B. Meixner, H. Kosch, and F. Lehner. "Challenges in creating multimedia instructions for support systems and dynamic problem-solving". In: *Proceedings of I-Know, 12th International Conference on Knowledge Management and Knowledge Technologies. i-KNOW '12*. New York, NY, USA: ACM, 2012, 33:1–33:4. DOI: 10.1145/2362456.2362497.

[Tsa+10] M.-F. Tsai, C.-K. Shieh, C.-H. Ke, and D.-J. Deng. "Sub-packet forward error correction mechanism for video streaming over wireless networks". In: *Multimedia Tools and Applications* 47.1 (2010), pp. 49–69. DOI: 10.1007/s11042-009-0406-5.

[Tsi+06] C. Tsinaraki, A. Perego, P. Polydoros, A. Syntzanaki, A. Martin, and S. Christodoulakis. “Semantic, constraint & preference based authoring of multi-topic multimedia presentations”. In: *Journal of Digital Information Management* 4.4 (2006). Website (accessed April 26, 2014), pp. 207–213. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.4052&rep=rep1&type=pdf.

[TR99] D. A. Turner and K. W. Ross. “Optimal streaming of synchronized presentations multimedia”. In: *Proceedings of the 7th ACM International Conference on Multimedia (Part 2). MULTIMEDIA ’99*. New York, NY, USA: ACM, 1999, pp. 91–93. DOI: 10.1145/319878.319903.

[UU03] U.S. Department of Health and Human Services (HHS) and U.S. General Services Administration (GSA). *Research-based Web design & usability guidelines*. Website (accessed May 06, 2013). 2003. URL: http://www.usability.gov/guidelines/guidelines_book.pdf.

[Urs+07a] M. Ursu, J. Cook, V Zsombori, and I. Kegel. “A Genre-Independent Approach to Authoring Interactive Screen Media Narratives”. In: *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*. Website (accessed April 26, 2014). Westin Arlington Gateway, Virginia, USA: AAAI Press, 2007, pp. 173–180. URL: http://www.aaai.org/Papers/Symposia/Fall/2007/FS-07-05/FS07-05-029.pdf.

[Urs+07b] M. F. Ursu, J. J. Cook, V. Zsombori, R. Zimmer, I. Kegel, D. Williams, M. Thomas, J. Wyver, and H. Mayer. “Conceiving ShapeShifting TV: a computational language for truly-interactive TV”. In: *Interactive TV: a shared experience, 5th European Conference, EuroITV 2007*. Ed. by P. Cesar, K. Chorianopoulos, and J. F. Jensen. Vol. 4471. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer Berlin Heidelberg, 2007, pp. 96–106. DOI: 10.1007/978-3-540-72559-6_11.

[Urs+08] M. Ursu, I. Kegel, D. Williams, M. Thomas, H. Mayer, V. Zsombori, M. Tuomola, H. Larsson, and J. Wyver. “ShapeShifting TV: interactive screen media narratives”. English. In: *Multimedia Systems* 14.2 (2008), pp. 115–132. DOI: 10.1007/s00530-008-0119-z.

[Use09] Userfocus ltd. *247 web usability guidelines*. Website (accessed April 26, 2014). 2009. URL: http://www.userfocus.co.uk/resources/guidelines.html.

[VID10] VIDDIX B.V. *VIDDIX. Mix video with the web*. Website (accessed April 26, 2014). 2010. URL: http://www.viddix.com/.

[VJM09] R. Vaisenberg, R. Jain, and S. Mehrotra. “SMPL, a specification based framework for the semantic structure, annotation and control of SMIL documents”. In: *11th IEEE Intl. Symposium on Multimedia, 2009. ISM ’09*. IEEE, 2009, pp. 533–539. DOI: 10.1109/ISM.2009.114.

[VA+04] S. Van Assche, F. Hendrickx, N. Oorts, and L. Nachtergaele. “Multi-channel publishing of interactive multimedia presentations”. In: *Computers & Graphics* 28.2 (2004), pp. 193–206. DOI: 10.1016/j.cag.2003.12.006.

[VRV12] D. C. Vasiliadis, G. E. Rizos, and C. Vassilakis. “Class-based weighted fair queuing scheduling on quad-priority Delta Networks”. In: *International Journal of Parallel, Emergent and Distributed Systems* 27.5 (2012), pp. 435–457. DOI: 10.1080/17445760.2012.712694.

[Vic11] M. Victorin. “Evaluation von Metadaten-Standards aus dem Multimediabereich im Hinblick auf die Umsetzung von interaktiven, nicht linearen Videos”. Senior thesis. Passau, Germany: Universität Passau, 2011.

[Vid12] VideoClix Technologies Inc. *VideoClix. Turning viewers into customers*. Website (accessed April 26, 2014). 2012. URL: http://www.videoclix.tv.

[Vil01] L. Villard. “Authoring transformations by direct manipulation for adaptable multimedia presentations”. In: *Proceedings of the 2001 ACM Symposium on Document Engineering. DocEng ’01*. New York, NY, USA: ACM, 2001, pp. 125–134. DOI: 10.1145/502204.502206.

[W3C02] W3C. *XHTML+SMIL profile*. Ed. by D. Newman, A. Patterson, and P. Schmitz. Website (accessed April 26, 2014). 2002. URL: http://www.w3.org/TR/XHTMLplusSMIL/.

[W3C03] W3C. *Extensible Markup Language (XML)*. Website (accessed April 26, 2014). 2003. URL: http://www.w3.org/XML/.

[W3C12] W3C. *Synchronized Multimedia (SMIL)*. Website (accessed April 26, 2014). 2012. URL: http://www.w3.org/AudioVideo/.

[W3C13a] W3C. *Cascading Style Sheets home page*. Website (accessed April 26, 2014). 2013. URL: http://www.w3.org/Style/CSS/.

[W3C13b] W3C. *HTML 5.1 - A vocabulary and associated APIs for HTML and XHTML (W3C working draft 29 October 2013)*. Ed. by R. Berjon, S. Faulkner, T. Leithead, E. D. Navara, E. O’Connor, S. Pfeiffer, and I. Hickson. Website (accessed April 26, 2014). 2013. URL: http://www.w3.org/TR/html51/.

[W3C13c] W3C. *HTML5 - A vocabulary and associated APIs for HTML and XHTML (W3C candidate recommendation 6 August 2013)*. Ed. by R. Berjon, S. Faulkner, T. Leithead, E. D. Navara, E. O’Connor, S. Pfeiffer, and I. Hickson. Website (accessed April 26, 2014). 2013. URL: http://www.w3.org/TR/html5/Overview.html.

[W3C13d] W3C. *Scalable Vector Graphics (SVG)*. Website (accessed April 26, 2014). 2013. URL: http://www.w3.org/Graphics/SVG/.

[Wac13] F. Wackermann. “Ablauflogik in nicht-linearen, interaktiven Videos”. Senior thesis. Passau, Germany: Universität Passau, 2013.

[WWR95] T. Wahl, S. Wirag, and K. Rothermel. “TIEMPO: temporal modeling and authoring of interactive multimedia”. In: *Proceedings of the International Conference on Multimedia Computing and Systems, 1995*. IEEE, 1995, pp. 274–277. DOI: 10.1109/MMCS.1995.484933.

[WLS11] J. Waitelonis, N. Ludwig, and H. Sack. “Use what you have: Yovisto video search engine takes a semantic turn”. In: *Semantic Multimedia*. Ed. by T. Declerck, M. Granitzer, M. Grzegorzek, M. Romanelli, S. Rüger, and M. Sintek. Vol. 6725. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 173–185. DOI: 10.1007/978-3-642-23017-2_12.

[Wan+02] B. Wang, S. Sen, M. Adler, and D. Towsley. “Optimal proxy cache allocation for efficient streaming media distribution”. In: *INFOCOM 2002. 21st Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*. Vol. 3. IEEE, 2002, pp. 1726–1735. DOI: 10.1109/INFCOM.2002.1019426.

[Wan+01] C. Wang, K. Long, X. Gong, and S. Cheng. “SWFQ: a simple weighted fair queueing scheduling algorithm for high-speed packet switched network”. In: *IEEE International Conference on Communications, 2001. ICC 2001*. Vol. 8. IEEE, 2001, pp. 2343–2347. DOI: 10.1109/ICC.2001.936551.

[WWL02] S. Wang, Y.-C. Wang, and K.-J. Lin. “Integrating priority with share in the priority-based weighted fair queuing scheduler for real-time networks”. In: *Real-Time Systems* 22.1-2 (2002), pp. 119–149. DOI: 10.1023/A:1013485520989.

[Wan+12] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang. “Propagation-based social-aware replication for social video contents”. In: *Proceedings of the 20th ACM International Conference on Multimedia. MM ’12*. New York, NY, USA: ACM, 2012, pp. 29–38. DOI: 10.1145/2393347.2393359.

[Wat87] J. A. Watlington. “Synthetic movies”. Website (accessed April 26, 2014). Master’s thesis. Massachusetts Institute of Technology, 1987. URL: http://alumni.media.mit.edu/~wad/ms-thesis/ms-thesis.pdf.

[Wau+06] T. Wauters, W. Van de Meerssche, F. De Turck, B. Dhoedt, and P. Demeester. “Co-operative proxy caching algorithms for time-shifted IPTV services”. In: *32nd EUROMICRO Conference on Software Engineering and Advanced Applications, 2006. SEAA ’06.* IEEE, 2006, pp. 379–386. DOI: 10.1109/EUROMICRO.2006.29.

[Wei10] A. Weick. “Evaluation von RIA-Technologien und Implementierung eines Videoplayers zur Wiedergabe interaktiver Videos”. Bachelor thesis. Passau, Germany: Universität Passau, 2010.

[Wei12] A. Weinast. “Kollaboration in interaktiven Videos - Konzeption und Implementierung”. Senior thesis. Passau, Germany: Universität Passau, 2012.

[Wes95] D. Wessels. “Intelligent caching for World-Wide Web objects”. Website (accessed April 26, 2014). PhD thesis. Boulder, CO, USA: University of Colorado at Boulder, 1995. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.6579&rep=rep1&type=pdf.

[Wha+94] C. Wharton, J. Rieman, C. Lewis, and P. Polson. “The cognitive walkthrough method: a practitioner’s guide”. In: *Usability inspection methods*. Ed. by J. Nielsen and R. L. Mack. New York, NY, USA: John Wiley & Sons, Ltd., 1994, pp. 105–140. URL: http://www.colorado.edu/ics/sites/default/files/attached-files/93-07.pdf.

[Wij+11] M. Wijnants, W. Vanmontfort, J. Dierckx, P. Quax, W. Lamotte, K. Moor, and J. Vanattenhoven. “Quality of service and quality of experience correlations in a location-based mobile multiplayer role-playing game”. In: *Entertainment Computing - ICEC 2011*. Ed. by J. C. Anacleto, S. Fels, N. Graham, B. Kapralos, M. Saif El-Nasr, and K. Stanley. Vol. 6972. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 101–112. DOI: 10.1007/978-3-642-24500-8_11.

[Wir12] WireWax ltd. *wireWAX*. Website (accessed April 26, 2014). 2012. URL: http://www.wirewax.com.

[WFW96] K. H. Wolf, K. Froitzheim, and M. Weber. “Interactive video and remote control via the World Wide Web”. In: *Interactive Distributed Multimedia Systems and Services*. Ed. by B. Butscher, E. Moeller, and H. Pusch. Vol. 1045. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1996, pp. 91–104. DOI: 10.1007/3-540-60938-5_7.

[WKD96] J. Wong, S. Kini, and K. Doobagunta. “Synchronization in specification-based multimedia presentations”. In: *Software: Practice and Experience* 26.1 (1996), pp. 71–81. DOI: 10.1002/(SICI)1097-024X(199601)26:1<71::AID-SPE997>3.0.CO;2-R.

[WRR97] J. Wong, S. Rao, and N. Ramaiah. “A multimedia presentation toolkit for the World Wide Web”. In: *Software: Practice and Experience* 27.4 (1997), pp. 425–446. DOI: 10.1002/(SICI)1097-024X(199704)27:4<425::AID-SPE92>3.0.CO;2-2.

[Won06] K.-Y. Wong. “Web cache replacement policies: a pragmatic approach”. In: *IEEE Network* 20.1 (2006), pp. 28–34. DOI: 10.1109/MNET.2006.1580916.

[Won+07] Y. W. Wong, J. Y. B. Lee, V. O.-K. Li, and G. S. H. Chan. “Supporting interactive video-on-demand with adaptive multicast streaming”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.2 (2007), pp. 129–142. DOI: 10.1109/TCSVT.2006.888839.

[WA97] R. P. Wooster and M. Abrams. “Proxy caching that estimates page load delays”. In: *Computer Networks and ISDN Systems* 29.8-13 (1997), pp. 977–986. DOI: 10.1016/S0169-7552(97)00041-X.

[Wor09] C. Worthington. *Basics film-making 01: producing*. illustrated. AVA Publishing, 2009.

[WCG06] G. Wu, E. K. P. Chong, and R. Givan. “Predictive buffer control in delivering remotely stored video using proxy servers”. In: *Computer Networks* 50.18 (2006), pp. 3721 –3742. DOI: 10.1016/j.comnet.2006.04.004.

[XZZ03] Z. Xiang, Q. Zhang, and W. Zhu. “Cache replacement and server selection for video proxy across wireless Internet”. In: *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS ’03*. Vol. 2. IEEE, 2003, pp. II–828 –II–831. DOI: 10.1109/ISCAS.2003.1206102.

[XCL11] X. Xiu, G. Cheung, and J. Liang. “Frame structure optimization for interactive multiview video streaming with bounded network delay”. In: *2011 18th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2011, pp. 593–596. DOI: 10.1109/ICIP.2011.6116619.

[XCL12] X. Xiu, G. Cheung, and J. Liang. “Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis”. In: *IEEE Transactions on Multimedia* 14.4 (2012), pp. 1109–1126. DOI: 10.1109/TMM.2012.2191267.

[YCW04a] C.-C. Yang, C.-K. Chu, and Y.-C. Wang. “Dividable dynamic timeline-based authoring for SMIL 2.0 presentations”. In: *2004 IEEE International Conference on Multimedia and Expo, 2004. ICME ’04*. Vol. 2. IEEE, 2004, pp. 1243–1246. DOI: 10.1109/ICME.2004.1394447.

[YCW04b] C.-C. Yang, C.-K. Chu, and Y.-C. Wang. “Reuse of SMIL 2.0 scripts in dividable dynamic timeline-based authoring”. In: *2004 IEEE International Conference on Multimedia and Expo, 2004. ICME ’04*. Vol. 2. IEEE, 2004, pp. 1235–1238. DOI: 10.1109/ICME.2004.1394445.

[YCW08] C.-C. Yang, C.-K. Chu, and Y.-C. Wang. “Extension of timeline-based editing for non-deterministic temporal behavior in SMIL2.0 authoring”. In: *Journal of Information Science and Engineering* 24.5 (2008). Website (accessed April 26, 2014), pp. 1377–1395. URL: http://erdos.csie.ncnu.edu.tw/~ccyang/Publication/JISE2008-3.pdf.

[YY03] C.-C. Yang and Y.-Z. Yang. “SMILAuthor: an authoring system for SMIL-based multimedia presentations”. In: *Multimedia Tools and Applications* 21.3 (2003), pp. 243–260. DOI: 10.1023/A:1025770817293.

[YZZ01] Q. Yang, H. H. Zhang, and H. Zhang. “Taylor series prediction: a cache replacement policy based on second-order trend analysis”. In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001*. 2001, pp. 1–7. DOI: 10.1109/HICSS.2001.926537.

[YKS99] T. Yatabe, H. Kawasaki, and M. Sakauchi. “Interactive video description on the network-interactive video representation of real world based on digital city map”. In: *IEEE International Conference on Multimedia Computing and Systems, 1999*. Vol. 2. IEEE, 1999, pp. 194–198. DOI: 10.1109/MMCS.1999.778255.

[YYL96] M. Yeung, B.-L. Yeo, and B. Liu. “Extracting story units from long programs for video browsing and navigation”. In: *Proceedings of the 3rd IEEE International Conference on Multimedia Computing and Systems, 1996*. IEEE, 1996, pp. 296–305. DOI: 10.1109/MMCS.1996.534991.

[You13] YouTube, LLC. *YouTube video annotations*. Website (accessed April 26, 2014). 2013. URL: http://www.youtube.com/t/annotations_about.

[Yue+11] G. Yue, N. Wei, J. Liu, X. Xiong, and L. Xie. “Survey on scheduling technologies of P2P media streaming”. In: *Journal of Networks* 6.8 (2011), pp. 1129–1136. DOI: 10.4304/jnw.6.8.1129-1136.

[ZSB02] C. Zahn, S. Schwan, and B. Barquero. “Authoring hypervideos: design for learning and learning by design”. In: *Writing Hypertext and Learning*. Ed. by R. Bromme and E. Stahl. London: Pergamon Press, 2002, pp. 153–176.

[Zee13] Zeega. *ZEEGA*. Website (accessed April 26, 2014). 2013. URL: http://zeega.com/.

[ZJ98] A. Zhang and T. V. Johnson. “Scheduling multimedia presentations in educational digital libraries”. In: *International Journal on Digital Libraries* 1.4 (1998), pp. 386–400. DOI: 10.1007/s007990050031.

[Zha+06] D. Zhang, L. Zhou, R. O. Briggs, and J. F. Nunamaker. “Instructional video in e-learning: assessing the impact of interactive video on learning effectiveness”. In: *Information & Management* 43.1 (2006), pp. 15–27. DOI: 10.1016/j.im.2005.01.004.

[Zha90] L. Zhang. “Virtual clock: a new traffic control algorithm for packet switching networks”. In: *ACM SIGCOMM Computer Communication Review* 20.4 (1990), pp. 19–29. DOI: 10.1145/99517.99525.

[ZEV07] Y. Zhao, D. L. Eager, and M. K. Vernon. “Scalable on-demand streaming of nonlinear media”. In: *IEEE/ACM Transactions on Networking* 15.5 (2007), pp. 1149–1162. DOI: 10.1109/TNET.2007.896534.

[ZA05] B. Zheng and M. Atiquzzaman. “System design and network requirements for interactive multimedia”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 15.1 (2005), pp. 145–153. DOI: 10.1109/TCSVT.2004.839982.

[ZA03] B. Zheng and M. Atiquzzaman. “Interactive video over ATM: state of the art”. In: *Computer Communications* 26.7 (2003), pp. 747–758. DOI: 10.1016/S0140-3664(02)00209-8.

[ZGJ05] T. T. Zhou, T. Gedeon, and J. S. Jin. “Automatic generating detail-on-demand hypervideo using MPEG-7 and SMIL”. In: *Proceedings of the 13th Annual ACM International Conference on Multimedia. MULTIMEDIA ’05*. New York, NY, USA: ACM, 2005, pp. 379–382. DOI: 10.1145/1101149.1101230.

[ZK99] D. Zotkin and P. J. Keleher. “Job-length estimation and performance in backfilling schedulers”. In: *The Eighth International Symposium on High Performance Distributed Computing, 1999. Proceedings*. IEEE, 1999, pp. 236–243. DOI: 10.1109/HPDC.1999.805303.

[Zwi12] S. Zwicklbauer. “Automatische Szenenerkennung mit manueller Korrekturmöglichkeit - Konzeption, Implementierung”. Master’s thesis. Passau, Germany: Universität Passau, 2012.

[ZMK14] S. Zwicklbauer, B. Meixner, and H. Kosch. “Improving scene detection algorithms using new similarity measures”. In: *MultiMedia Modeling*. Ed. by C. Gurrin, F. Hopfgartner, W. Hurst, H. Johansen, H. Lee, and N. O’Connor. Vol. 8326. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 323–330. DOI: 10.1007/978-3-319-04117-9_31.

[imc10] imc AG. *LECTURNITY 4*. Website (accessed April 26, 2014). 2010. URL: http://www.lecturnity.com/en/lecturnity/features/.

[mem13] memex Academy. *Riva Bundle “Autor”*. Website (accessed April 26, 2014). 2013. URL: http://memex-academy.eu/.

Index

Actions, 54
Additional information, 70
Algorithm
    GetStartFrame, 110
    SortSceneLinear, 110
    SortVideoLinear, 120
Annotated interactive non-linear video, 18, 94
    definition, 18, 94
Annotation editors, 62
Annotations
    SIVA Player, 70
    SIVA Producer, 62
Authoring tool, 11, 29, 223
    clickable video, 29, 231
    hypervideo, 32, 228
    interactive video, 30, 225
    multimedia presentation, 31, 232
    non-linear video, 30, 224
    related work, 29
Avoiding of deadlocks, 125

Bandwidth, 100
Button panel, 68

Cache control, 108
Cache management, 11, 75, 105
Cache management and replacement strategies, 88
Cache size, 99
Calculation of the starting point, 110
Categories of elements, 122
Choice at forks provider, 107
Cleaning agent, 108
Clickable video, 14, 29, 38, 231, 247
    authoring tool, 29
    definition, 14
    players, 38
Collaboration, 71
Combination of algorithms/strategies (patterns), 175
    delete, 178
    playback, 178
    summary, 179
    weighted metrics, 179
Communication architecture, 106
    cache control, 108
    choice at forks provider, 107
    cleaning agent, 108
    decoder, 108
    delete scheduler, 108
    download agent, 107
    download scheduler, 107
    internal structure, 107
    player, 108
    scenario handler, 108
    scene scheduler, 107
    UML sequence chart, 108, 253
    UML state chart, 109
    video control, 108
Conceptual model, 48
Control file, 146
CSS, 21

Deadlock, 125
    “Circular wait” condition, 125
    “Mutual exclusion” condition, 125
    “No preemption” condition, 125
    “Wait for” condition, 125
Decision at forks, 119
Decoder, 108
Definition
    annotated interactive non-linear video, 18, 94, 96
    annotation, 95
    bandwidth, 100
    cache size, 99
    clickable Video, 14
    content of an annotation, 95

DFA, 96
dimension function, 97
duration of a scene, 98
end scene, 95
extended interactivity and navigation, 101
frame, 95
frame rate, 97
hypervideo, 18
interactive video, 14
multimedia presentation, 17
non-linear video, 16
priority, 98
priority of an annotation, 95
projection function, 97, 98
rame rate, 97
scene, 95
set of all successor scenes of a scene, 99
set of annotations, 95
set of annotations of a scene, 98, 99
set of downloadable elements, 98
set of downloadable elements of a scene, 99
set of downloadable elements of a video $\mathcal{V}$, 98
set of frames, 95
set of frames of a scene, 98, 99
set of scenes, 95
size function, 98
start scene, 95
tuple of frames of a scene, 98
user behavior, 100
valid schedule for one scene, 115
VCR actions, 101
Delete scheduler, 108
Delete strategies, 120, 302
categories of elements, 122
delete threshold, 125
indices of deletion, 123
Description format, 19
DFA, 96
Dimension function, 97
Downlaod agent, 107
Downlaod scheduler, 107
Download, 75, 105
Download and cache management, 105
Download and streaming of interactive (non-linear) video, 84
Download scheduling, 11, 115
constraints, 115
decision at forks, 119
download strategy, 120
pre-fetch strategies, 117
Download strategy, 120
Duration of a scene, 98

Environment file, 145
Evaluation, 129, 287
Evaluation number of annotations, 181, 314
DLV metric, 186, 317
$DL_{not\ watched}$ metric, 184, 315
P_{sum} metric, 184, 315
$RDLV$ metric, 316
WF_{start} metric, 182, 314
WT_{start} metric, 182, 314
summary, 186
Evaluation of the delete strategies, 169
DLV metric, 173, 175, 314
$DL_{not\ watched}$ metric, 314
P_{sum} metric, 173, 314
$RDLV$ metric, 314
WF_{start} metric, 171, 302, 314
WT_{start} metric, 314
summary, 175
Evaluation of the pre-fetch strategies, 156
DLV metric, 167, 169, 302
$DL_{not\ watched}$ metric, 164, 165, 302
P_{sum} metric, 160, 162, 302
$RDLV$ metric, 165, 167, 302
WF_{start} metric, 157, 158, 290, 302
WT_{start} metric, 160, 302
summary, 169
Evaluation pattern width and probabilities, 187, 318
DLV metric, 320, 321
$DL_{not\ watched}$ metric, 188
P_{sum} metric, 319, 320
WF_{start} metric, 188, 318
WT_{start} metric, 318, 319
summary, 188
Evaluation priority-based strategies, 323
$A_{skipped}$ metric, 198
DLV metric, 198
$DL_{not\ watched}$ metric, 198

P_{sum} metric, 195
$RDLV$ metric, 198
WF_{start} metric, 195
WT_{start} metric, 195
Extended interactivity and navigation, 101

Formal definition, 10
Formalized video model, 10, 93
Frame rate, 97

GetStartFrame, 110
Global calculations, 110

Hardware constraints, 93, 99
Hotspot, 64, 72
HTML5, 21
Hypermedia applications, 24, 214
Hypervideo, 18, 32, 41, 215, 228, 244
authoring tool, 32
definition, 18
language, 26
model, 26

Indices of deletion, 123
Interactive multimedia presentation, 21, 217
Interactive video, 14, 30, 38, 225, 241
authoring tool, 30
definition, 14
players, 38
Interactivity
SIVA Player, 71
SIVA Producer, 64
Interactivity in linear videos, 77

JSON, 21

Keyword search, 59
Keywords, 64
Kruskal-Wallis
results, 279

Latency reduced streaming in hypervideos, 78

Mobile player, 11
Mulitmedia presentations, 39
Multi-view video, 13
Multimedia model, 19, 24
Multimedia presentation, 17, 31, 218, 232, 248
authoring tool, 31
definition, 17
languages, 21
models, 21
players, 39
Multiple comparison test, 279
DLV metric, 285
$DL_{not\ watched}$ metric, 284
P_{sum} metric, 283
$RDLV$ metric, 284
WF_{start} metric, 280
WT_{start} metric, 282

NCL, 23
Non-linear video, 16, 30, 39, 224, 240
authoring tool, 30
definition, 16
players, 39
Non-linearity
SIVA Player, 68
SIVA Producer, 61

PathSet file, 145
Performance evaluation, 148
combination of algorithms/strategies (patterns), 175
delete strategies, 169, 302
pattern width and probabilities, 318
pre-fetch strategies, 156, 290
priority-based strategies, 323
user generated scenarios, 190, 322
varying annotation priorities, 194
varying numbers of annotations, 181, 314
varying path probabilities, 187
Performance metrics, 130
$A_{skipped}$ metric, 132
DLV metric, 132
$DL_{not\ watched}$ metric, 132
P_{sum} metric, 131
$RDLV$ metric, 132
WF_{start} metric, 131
WT_{start} metric, 131
Player, 108
Player buttons, 71
Player implementations, 75
Players, 239
clickable video, 38, 247
hypervideo, 41, 244

interactive video, 38, 241
multimedia presentation, 39, 248
non-linear video, 39, 240
others, 38
related work, 38
Pre-fetch strategies, 117, 290
Priority, 98
Priority strategies
$A_{skipped}$ metric, 337
DLV metric, 337
$DL_{not\ watched}$ metric, 337
P_{sum} metric, 337
$RDLV$ metric, 337
WF_{start} metric, 323, 337
WT_{start} metric, 337
Problem statement, 7
authoring, 7
playback, 7
user experience, 8
Project information, 52
Projection function, 97, 98

Reference models, 25
Related work
authoring tools, 29
cache management, 75
cache management and replacement strategies, 88
clickable video, 29, 38
CSS, 21
description format, 19
download, 75
download and streaming of interactive (non-linear) video, 84
HTML5, 21
hypermedia applications, 24
hypervideo, 32, 41
hypervideo languages, 26
hypervideo models, 26
interactive video, 30, 38
interactivity in linear videos, 77
languages for interactive multimedia presentations, 21
latency reduced streaming in hypervideos, 78
models for interactive multimedia presentation, 21
multimedia model, 19
multimedia presentation, 31, 39
NCL, 23
non-linear video, 30, 39
other authoring tools, 29
other multimedia models, 24
player implementations, 75
players, 38
reference models, 25
scheduling of download queues, 80
SMIL, 21
standards, 19
SVG, 21
temporal synchronization, 25
user behavior during video playback, 77
Research contributions, 10
authoring tool, 11
cache management, 11
download, 11
download scheduling, 11
formal definition, 10
formalized video model, 10
mobile player, 11
simulation framework, 11
software suite, 10
user interaction, 11
web-player, 11
XML file, 10
Resources, 53

Scenario file, 145
Scenario handler, 108
Scenarios, 255
scenario A, 258
scenario B, 261
scenario C, 140
scenario D, 263
task description, 255
Scene graph, 62
Scene list, 57
Scene scheduler, 107
Scheduling of download queues, 80
Search, 69
Selection panel, 65
Set of all successor scenes of a scene, 99
Set of annotations of a scene, 98
Set of downloadable elements, 98
Set of frames of a scene, 98
Settings file, 145
Simulation environment, 144

control file, 146
environment file, 145
getpoolsize, 147
hardware, 148
implementation of the framework, 146
internal structure of the framework, 145
pathset file, 145
resume, 147
scenario file, 145
settings file, 145
simulate, 147
simulation settings, 145
simulation tool, 146
strategy file, 146
XML generator, 146
Simulation framework, 11
Simulation hardware, 148
Simulation settings, 145
Simulation tool, 146
SIVA Player, 66
additional information, 70
annotations, 70
button panel, 68
collaboration, 71
hotspot, 72
interactivity, 71
non-linearity, 68
player buttons, 71
quiz, 68
search, 69
table of contents, 69
SIVA Producer, 60
annotation editors, 62
annotations, 62
definition of keywords, 64
hotspots, 64
interactivity, 64
non-linearity, 61
scene graph, 62
selection panel, 65
table of contents, 62
SIVA Suite, 45
actions, 54
Android player app, 67
annotations, 47
conceptual model, 48
data exchange, 45
data flow, 45
interactivity, 47
iPhone app, 68
keyword search, 59
non-linearity, 47
project information, 52
resources, 53
scene list, 57
SIVA Player, 66
SIVA Producer, 60
table of contents, 58
XML file, 50
XML schema, 48, 50
Size function, 98
SMIL, 21
Social TV, 14
Social video, 14
Software suite, 10
Sorting elements of a scene, 110
SortSceneLinear, 110
SortVideoLinear, 120
Standard, 19
Standards/models
hypermedia applications, 214
hypervideos, 215
interactive multimedia presentations, 217
multimedia presentations, 218
Starting point, 110
Statistics, 149, 265, 287
DLV metric, 154, 155, 277
$DL_{not\ watched}$ metric, 154, 273
H-value, 149
P_{sum} metric, 152, 153, 271
$RDLV$ metric, 154, 275
WF_{start} metric, 150, 267
WT_{start} metric, 151, 152, 269
Kruskal-Wallis test, 149
multiple comparison test, 150
non-parametric test, 150
post hoc test, 150
Strategy
DELETE_D_PROB, 124
DELETE_LRU, 124
DELETE_PRIO, 124
DELETE_SD, 123
PLAY_MIN_REL_PRIO($f_{m/n}, \Lambda$), 115
PLAY_MIN_REL(f_m), 115

PLAY_SCENE, 114
PLAY_STARTUP(f_x), 115
PREFETCH_FF, 118
PREFETCH_SL, 117
Strategy file, 146
SVG, 21

Table of contents, 58, 62, 69
Temporal synchronization, 25
Test configurations, 136
cycle pattern, 133
description of the video, 136
mirrorworld/counterpoint pattern, 134
patterns, 137
scenarios, 140
sequence pattern, 133
sieve pattern, 134
split/join pattern, 133
user behavior, 136, 140
Test graph, 132, 134
Test pattern, 132, 133
Tuple of frames of a scene, 98

UML sequence chart, 108, 253
UML state chart, 109
Use cases, 4
e-learning, 5
interactive video stories, 4
memory training, 7
mobile help systems, 6
sports videos, 4
virtual tours, 4
User behavior, 93, 100
User behavior during video playback, 77
User generated scenarios, 322
DLV metric, 191
$DL_{not\ watched}$ metric, 322
P_{sum} metric, 191
$RDLV$ metric, 322
WF_{start} metric, 191
WT_{start} metric, 191
summary, 191
User interaction, 11

Valid schedule for one scene, 115
VCR actions, 101
Video annotation, 14
Video browsing, 13
Video control, 108
Video dependent limitations, 93
Video model, 93
Video playback, 114
Video search, 13

Web-player, 11

XML, 21
XML file, 10
XML generator, 146
XML schema, 48
XML schema/XML file, 50
actions, 54
keyword search, 59
project information, 52
resource, 53
scene list, 57
table of contents, 58