

Workshop: DSpace und das Semantic Web

Agenda

- Ziel & Motivation
- Kurze Linked Data Einführung
- Repositorien und das Semantic Web
- dspace-rdf: Aufbau und Konfiguration
- Questions & Answers
- Evtl. Live-Demonstration

**Workshop-Charakter: Fragen, Ideen, Anmerkungen bitte jederzeit!
Konkrete Konfiguration am Ende.**

Kurze Vorstellung der Teilnehmer: Wer? Woher? Welche Version von DSpace?

Ziel & Motivation

Digitale Repositorien

Ein Repository ist ein System zur sicheren Speicherung digitaler Objekte und der die Objekte beschreibenden strukturierten Metadaten.

Beispiele:

- Digitale Archive
- Institutionelle Repositorien (Preprints, Postprints, OA-Publikationen, ...)
- Digitale Bilddatenbanken
- Forschungsdaten-Repositorien
- ...



Mehr als 2500 OA-Repositorien weltweit

Source: The Directory of Open Access Repositories, <http://www.opendoar.org>, abgerufen am 06.06.2014.

Verbreitung von Repositoriensoftware

	OpenDOAR		ROAR	
DSpace	1127	41,2%	1464	39,2%
EPrints	375	13,7%	532	14,2%
OPUS	71	2,6%	54	1,4%
Fedora	41	1,5%	55	1,5%
MyCore	8	0,3%	7	0,2%
Andere	835	30,5%	516	13,8%
Unbekannt	282	10,3%	1111	29,7%
	2739		3739	

Quelle: <http://www.opendoar.org> und <http://roar.eprints.org>,
abgerufen am 06. Juni 2014.

xxx.lanl.org / ArXiv.org

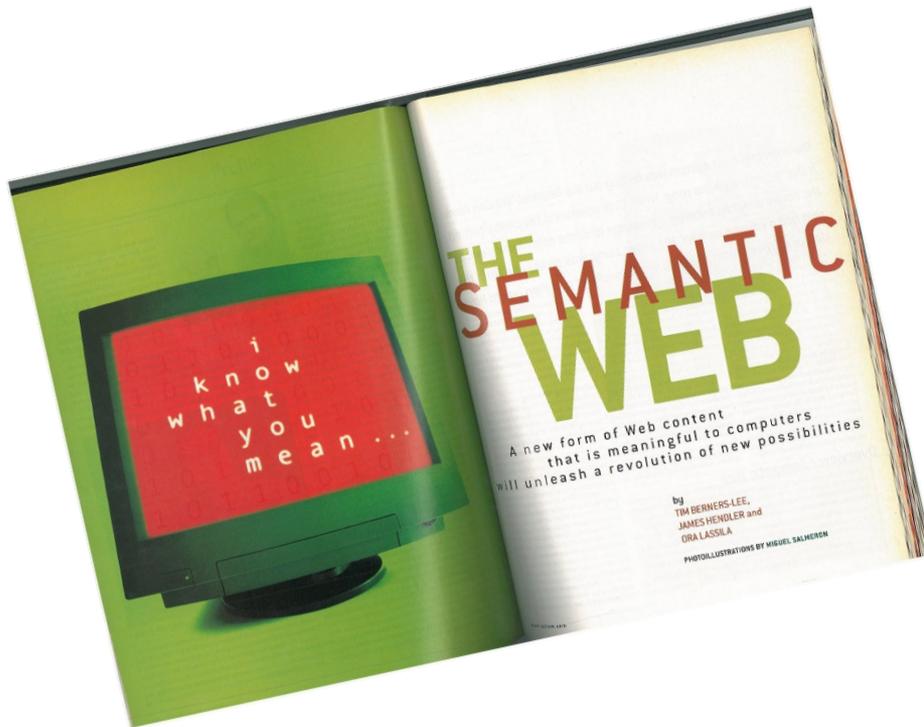


“Although the WorldWideWeb still represents only a small fraction of the overall usage, this access mode is expected to become dominant in the near future.”

Paul Ginsparg 1994

Quelle: Paul Ginsparg, *First Steps Towards Electronic Research Communication*. In: *Computer in Physics*, Vol. 8, No. 4, 1994, pp. 390-396.
Foto: Paul Ginsparg

The Semantic Web



„Information varies along many axes. One of these is the difference between information produced primarily for human consumption and that produced mainly for machines. [...]

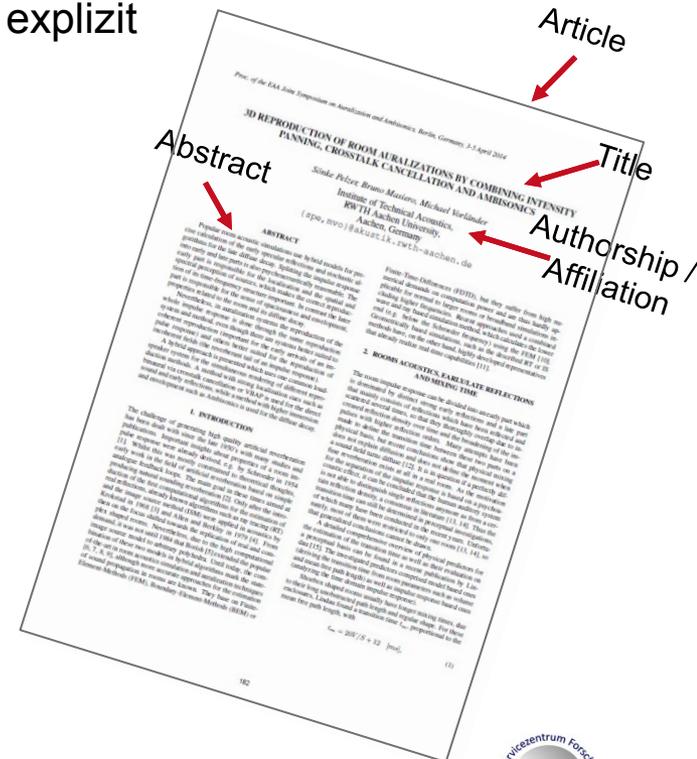
To date, the Web has developed most rapidly as a medium of documents for people rather than for data and information that can be processed automatically.“

Berners-Lee, Handler, Lasilla 2001

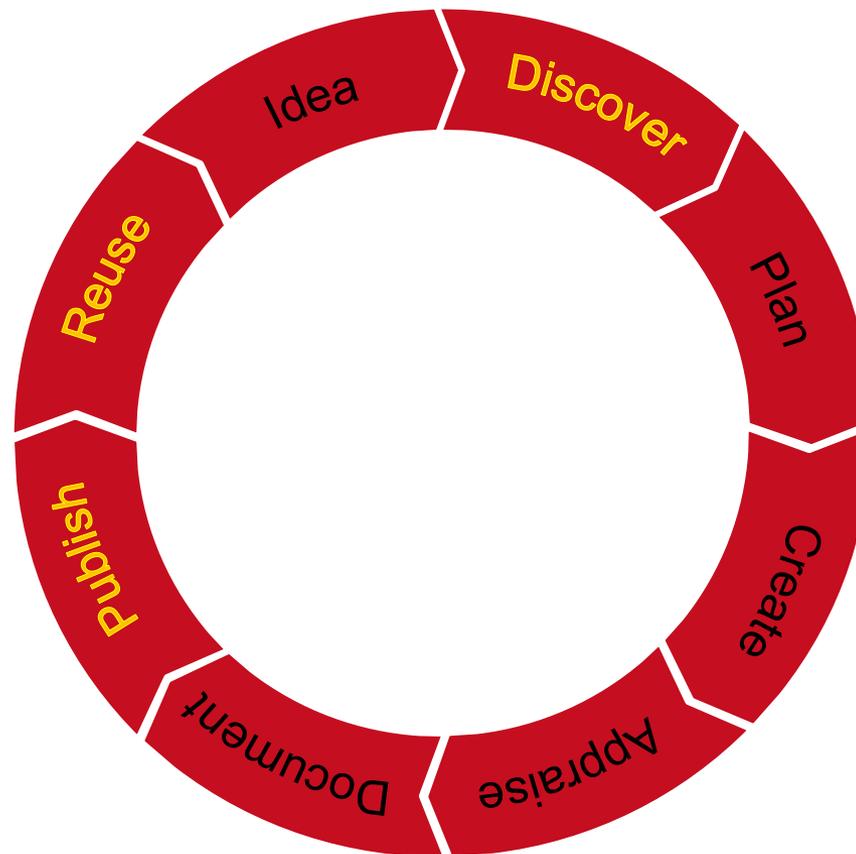
Repositorieninhalte und das Semantic Web

- Informationen im Internet sind auf den menschlichen Konsum ausgerichtet
- Viele Informationen werden nur implizit über den Kontext zur Verfügung gestellt
- Grundidee des Semantic Web: Implizite Informationen explizit machen, um sie automatisiert verarbeiten zu können

- Ein Repository „kennt“ den Titel eines Artikels
- Titel werden Fett und in größerer Schrift ausgegeben, Autorennamen kursiv
- Wieso stellen wir diese Daten nicht auch maschinell nutzbar bereit?



Research Data Lifecycle



Bisheriger Datenaustausch mit Repositorien

- OAI-PMH (Open Archive Initiative – Protocol for Metadata Harvesting):
de-facto-Standard im Repositorienumfeld
 - Aber auf das Repositorienumfeld begrenzt!
 - Google unterstützt OAI-PMH seit 2008 nicht mehr
(zuvor als Alternative zum Sitemap Protocol genutzt)
 - „Nur“ eine Schnittstelle, kein Format
-
- Linked Data ist ein generischer nativer Weg Daten auszutauschen
 - Linked Data ist nicht auf das Umfeld von Repositorien beschränkt
 - Linked Data ist selbst beschreibend
 - **Linked Data vereinfacht den Datenaustausch mit Repositorien**

Linked Data – Kurze Einführung

The Linked Data Principles

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those things
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs, so they can discover more things

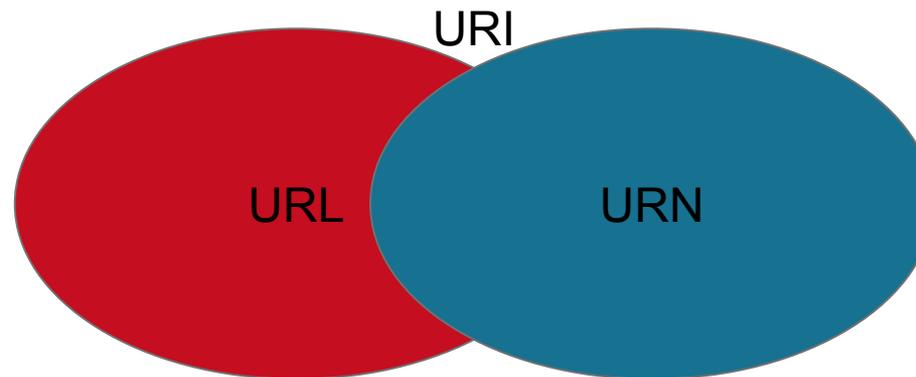
Tim Berners-Lee

<http://www.w3.org/DesignIssues/LinkedData.html>

- “This creates what I call a *Semantic Web* – a web of data that can be processed directly or indirectly by machines.”

Tim Berners-Lee und Marc Fischetti:
“Weaving the Web. The Past, Present and Future of the World Wide Web by its Initiator”
London 1999, ISBN: 0-7528-2090-7

URI = URN + URL



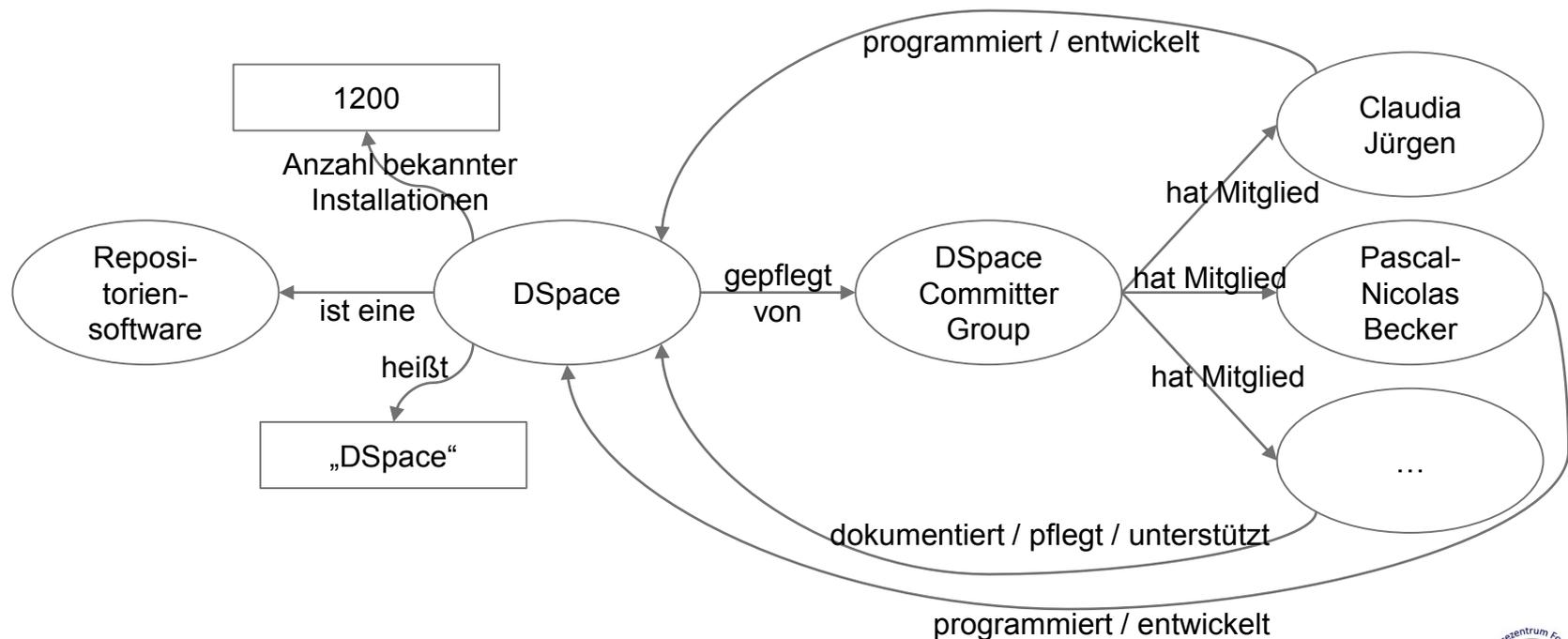
<http://digital-repositories.org/ontologies/dspace#Repository>
<mailto:p.becker@tu-berlin.de>

- Uniformed Resource Identifier ist ein Identifikator für eine abstrakte oder physische Ressource
- Ein URI kann ein Uniformed Resource Name, Uniformed Resource Locator oder beides sein
- Ein URN ist ein Name, ein URL ist eine Adresse
- Ein URI ist eindeutig!

RFC 3986

Resource Description Framework – RDF

- Datenmodell
- Formulierung logischer Aussagen über beliebige Ressourcen
- Struktur von Trippeln: Subject – Predicate – Object
- Subject und Predicate sind immer URIs, Objects sind URIs oder Literale



Vokabulare und Ontologien

- Zur Formulierung von Aussagen in RDF brauchen wir URIs
- URIs sollen eine „Bedeutung“ (\Rightarrow Semantik) haben
- Vokabulare dienen dazu URIs eine Semantik zuzuweisen

- Aus logischen Aussagen kann man Schlussfolgerungen ziehen
- Zum Teil ist Wissen erforderlich, das über die reinen Vokabulare hinausgeht
- Hintergrund- oder Domänenwissen wird in Ontologien abgelegt

- Ontologien und Vokabulare werden als Linked Data und somit in RDF veröffentlicht
- RDF-Schema (RDFS) dient der Beschreibung von Vokabularen
 - Klassenmodell, bei dem Klassen und Eigenschaften getrennt modelliert werden
- Die Web Ontology Language (OWL) geht über die Möglichkeiten von RDFS hinaus
 - für Ontologien genutzt

Beispiel: DSpace Metadata RDF Mapping Vocabulary

```

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
@prefix :    <http://digital-repositories.org/ontologies/dspace-metadata-mapping/0.2.0#> .
...
<http://digital-repositories.org/ontologies/dspace-metadata-mapping/0.2.0>
  rdfs:comment "Vocabulary for describing mappings of DSpace metadata to rdf." ;
  dc:contributor "Pascal-Nicolas Becker" ;
  dc:title      "DSpace Metadata RDF Mapping Spec" ;
  dc:description "Vocabulary for describing mappings of DSpace metadata to RDF. This
                 vocabulary is used to configure DSpace how to convert stored metadata
                 into RDF." ;
  dc:date       "2014-04-18".

:DSpaceMetadataRDFMapping
  a          rdfs:Class ;
  rdfs:label "DSpace Metadata RDF Mapping" ;
  rdfs:comment "Represents the mapping of a DSpace metadata value to an RDF equivalent.".

:Result
  a          rdfs:Class ;
  rdfs:subClassOf rdf:Statement ;
  rdfs:label  "DSpace Metadata RDF Mapping Result" ;
  rdfs:comment "A reified statement that describes the result of the
              DSpaceMetadataRDFMapping.".
...

```

Beispiel: DSpace Repository Ontology

```
@prefix : <http://digital-repositories.org/ontologies/dspace/#> .
```

```
...
```

```
<http://digital-repositories.org/ontologies/dspace/> rdf:type owl:Ontology ;
  rdfs:label "DSpace Repository Ontology"@en ;
  dc:creator "Pascal-Nicolas Becker" ;
  dc:date "2014-09-11" ;
  dc:description "DSpace Repository Ontology"@en ;
  dc:description "Ontology to describe a repository using DSpace. You can find further
    information about dspace at http://www.dspace.org."@en .
```

```
...
```

```
:hasBitstream rdf:type owl:ObjectProperty ;
  rdfs:comment "Links from an item to a bitstream of the item. Bitstreams can be a
    representation of an item or a part of a representation composed of several bitstreams.
    Bitstreams are arbitrary files, e.g. documents, archives, images, ..."@en ;
  rdfs:domain :Item ;
  rdfs:subPropertyOf :hasPart .
```

```
...
```

```
### http://digital-repositories.org/ontologies/dspace/#hasPart
:hasPart rdf:type owl:ObjectProperty ,
  owl:TransitiveProperty ;
  rdfs:comment "Links top down between the structure of a DSpace repository."@en ;
  owl:inverseOf :isPartOf .
```

```
...
```

Zusammenfassung

- RDF ist ein Datenmodell zur Formulierung logischer Aussagen
- Subject und Predicate sind immer URIs Objects sind URIs oder Literale
- URIs sind eindeutig („Johann Jakob Freudenreich ist Bräutigam und braut zu gleich“)
- Vokabulare und Ontologien sollen die Semantik einer URI beschreiben und Wissen „hinterlegen“
- Aus logischen Aussagen lassen sich Schlüsse ziehen
 - DSpace wird von der DSpace Committer Group gepflegt
 - Pascal-Nicolas Becker ist Mitglieder der DSpace Committer Group
 - Pascal-Nicolas Becker ist für die Pflege von DSpace (mit-)verantwortlich
- Ein Computer „versteh“ die Semantik nicht
- Ein Computer kann Logik anwenden
- Logische Schlussfolgerungen wirken wie Verständnis

RDF-Repräsentationen

- RDF ist ein Modell
- RDF ist unabhängig von einer speziellen Repräsentation
- Für RDF gibt es unterschiedliche Repräsentationen (Graphisch, RDF/XML, Turtle, N3, ...)
 - Turtle gilt als benutzerfreundliche Alternative zu RDF/XML und ist eine Teilmenge von N3
 - Die Default-Konfiguration von DSpace-RDF verwendet Turtle, die Konfiguration kann aber auch in anderen Repräsentationen angegeben werden
 - Die Beispiele in dieser Präsentation verwenden Turtle

SPARQL

- SPARQL Protocol And RDF Query Language ist eine Abfragesprache für RDF
- „Was SQL für relationale Datenbanken ist SPARQL für RDF“
- „SPARQL-Endpoint“: Schnittstelle, die über SPARQL abgefragt werden kann
- Seit SPARQL 1.1 ist auch die Manipulation von Daten via SPARQL möglich

```
SELECT * WHERE { ?s ?p ?o . }
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT ?s WHERE { ?s dc:contributor "Becker, Pascal-Nicolas" . }
```

SPARQL by example:

<http://www.cambridgesemantics.com/semantic-university/sparql-by-example>

Von RDF zu Linked Data

- RDF ist ein Datenmodell
- Linked Data ist ein Begriff für bestimmte Konventionen Daten zu bereitzustellen

Linked Data Principles:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those things
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs, so they can discover more things

Tim Berners-Lee

<http://www.w3.org/DesignIssues/LinkedData.html>

HTTP

- Linked Data setzt auf HTTP auf
- Die in RDF verwandten URIs sollen via HTTP aufrufbar sein und weitere Informationen liefern => **Linked Data**
- HTTP verfügt über Mechanismen um URLs weiterzuleiten
- Linked Data unterscheidet zwischen URNs und URLs: URNs sollen auf URLs weitergeleitet werden, die die Daten (RDF/XML, Turtle, ...) enthalten, die die durch den URN identifizierte Resource beschreiben
(<http://example.org/rdf/dog> => <http://example.org/dog.ttl>)
- doi:10.0123/4567 ist kein HTTP-URI! (dazu gleich mehr)

Vermeidbare Fehler und Hindernisse

- URIs nachnutzen, wo immer möglich! Keine eigenen URIs erzeugen, wenn es bereits adäquate gibt
 - Beispiele: Für Klassifikationen, wie die Library of Congress Subject Headings (LCSH) oder die Dewey Decimal Classification (DDC), existieren bereits URIs
- Erzeugung von URIs nur für „eigene“ Entitäten, Erzeugung von URIs nur wenn ausreichend Informationen vorhanden sind
- Beispiel: Autoren
 - Können verschiedene Autoren gleichen Namens differenziert werden?
 - Sollten Autoren nicht eigene URIs erzeugen und hinterlegen können, anstatt von „fremden“ Institutionen URIs zugewiesen zubekommen?
- Aber: Erzeugung von URIs für Objekte des Repositoriums (da autoritativ)
- Erzeugung von Links so oft wie möglich

Repositorien und das Semantic Web

Repositorieninhalte sind besonders gut geeignet



Inhalte von Repositorien sind zur Bereitstellung im Semantik Web besonders gut geeignet:

- Metadaten existieren bereits in einer strukturierten Form
- Sie müssen nicht erst generiert oder aufwändig manuell zur Publikation als Linked Data erzeugt werden
- Es geht "lediglich" um die Konvertierung vorhandener Daten in RDF, um das Hinzufügen von Links und das Publizieren unter Berücksichtigung der Linked Data Principles

Was speichern Repositorien?

“Ein Repository ist ein System zur sicheren Speicherung *und Weitergabe* digitaler Objekte und der die Objekte beschreibenden Metadaten.“

- Digitale Objekte
 - Eine oder mehrere Dateien: Dokumente (PDF, Text, ...), Bilder (PNG, Tiff, ...), Tabellen (CSV, ...), Audio (Wave, OGG/VORBIS, ...), Video, ...
 - Beschreibende Metadaten
 - Strukturierte Metadaten in Key-Value-Form: dc.title, dc.contributor.author, dc.description, dc.date.available, dc.subject.lcsh, dc.subject.ddc, ...
- Unmöglich die Dateien zu konvertieren (technische Probleme, manuell nicht leistbar)
- Aber wir können die Metadaten konvertieren und die Dateien verlinken!

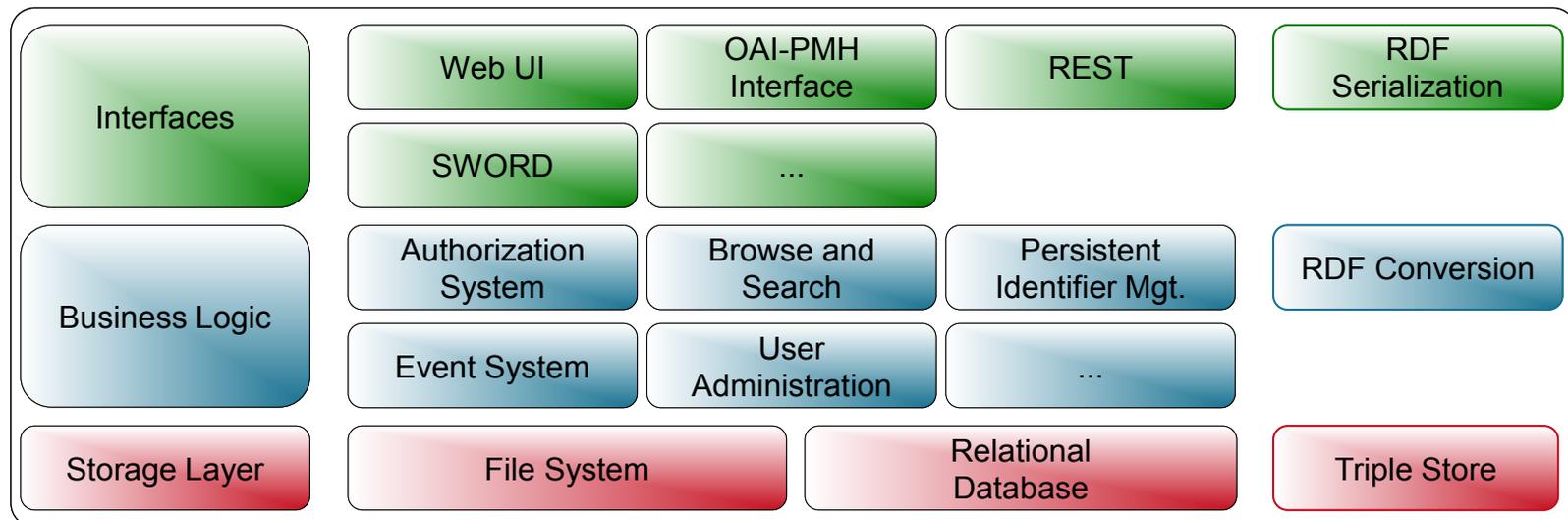
Full metadata record		
DC Field	Value	Language
dc.contributor.author	Lindau, Alexander	-
dc.date.accessioned	2014-02-27T14:19:30Z	-
dc.date.available	2014-02-27T14:19:30Z	-
dc.date.issued	2014-02-27	-
dc.identifier.uri	http://depositonce.tu-berlin.de/handle/11303/157	-
dc.identifier.uri	http://dx.doi.org/10.14279/depositonce-1	-
dc.description	The 'SAQI. Test Manual. v1.0' documents the complete German and English version of the SAQI. It serves as an user-oriented introduction giving valuable hints for practical application, e.g., by referring to the whisper Matlab toolbox v1.8.0 which features a full implementation of a SAQI test. Additional resources are provided in a zip-container. It includes relevant project-related publications, illustrative audio examples, empirical test data sets, and Matlab functions for convenient later statistical analysis and plotting of SAQI test results. Folder structure in 'SAQI. Test Manual. v1.0. Additional files.zip': /1 references /2 audio files /3 mfiles Further related resources: http://www.ak.tu-berlin.de/saqi http://www.ak.tu-berlin.de/whisper	en_US

Charakteristika von Repositorien

- Verschiedene Repositorien nutzen unterschiedliche Metadatenschemata
 - Die Konvertierung muss stark konfigurierbar und erweiterbar sein
- Für Metadatenschemata können bereits Vokabulare existieren (z.B. DC, DDC, LCSH, ...)
 - Werte von Metadaten müssen in URIs/Links konvertiert werden können
- Repositorieninhalte ändern sich so gut wie nie (Zitierfähigkeit, Zuverlässigkeit), die Konvertierung kann aber zeitaufwändig sein
 - Daten konvertieren und in einem Cache speichern
- Repositorien erzeugen URIs zur Adressierung ihrer Inhalte
 - Bestehende URIs nachnutzen und um Content Negotiation erweitern
- Persistent Identifier (handle, DOI, ...) verstoßen gegen die Linked Data Principles
 - Persistent Identifier sind in der Form als HTTP(S) URIs zu nutzen (z.B. <http://hdl.handle.net/...>, <http://dx.doi.org/...>)

Repositoriensoftware erweitern

- Hinzufügen eines Triple Stores (Triple Store = Datenbank für RDF)
 - Nutzung als Cache für die konvertierten Daten
 - Zur Bereitstellung eines SPARQL-Endpoints
- Methoden entwickeln zur Konvertierung in RDF und zur Ergänzung von Links
- Ein Modul zur Bereitstellung der konvertierten Daten als RDF-Serialisierung erstellen
- Content Negotiation unterstützen



Konvertierung vorhandener Metadaten in RDF

- Repositoriensoftware kann um weitere Metadatenfelder erweitert werden
- Dublin Core wird oft genutzt, auch andere Metadatenschemata müssen unterstützt werden
- Die Konvertierung muss stark konfigurierbar sein!
- Die Konfiguration sollte in RDF vorgenommen werden, da so die volle Funktionalität von RDF genutzt werden kann
- „Reification“ kann zur Beschreibung der gewünschten Ergebnisse genutzt werden
- Platzhalter können z.B. für URIs des Repositoriums verwendet werden
- Reguläre Ausdrücke können Literale und/oder URIs aus Metadatenwerten generieren
- Es braucht ein Vokabular um solche Konfigurationen zu erstellen

Repositorien beschreiben

- Neben der Konvertierung der Metadaten sollten die Repositorien selbst beschrieben werden
- Wer betreibt das Repository? Gibt es eine SWORD-Schnittstelle? Wo ist der SPARQL-Endpoint zu finden? Wie ist das Repository strukturiert? ...
- Auch braucht es ein Vokabular, um die Digitalen Objekte (Dateien) zu verlinken
- Für DSpace habe ich die DSpace Repository Ontology erstellt:
<http://digital-repositories.org/ontologies/dspace>
- Es wäre wichtig eine Digital Repositories Ontology zu erstellen, unabhängig von der genutzten Repositoriensoftware
- Ein Mapping zwischen einer solchen Ontologie und der DSpace Repository Ontology, der Eprints Ontology, ... wäre hilfreich! (Mitstreiter gesucht)

DSpace-RDF

DSpace 5

- DSpace ist die weltweit am häufigsten eingesetzte Software für OA-Repositoryn
- DSpace 5.0 wurde im Januar 2015 veröffentlicht, 5.2 ist im Mai erschienen
- Die TU Berlin hat DSpace 5 um Linked Data Support erweitert:
 - Konvertierung der Metadaten in Linked Data
 - Ausgabe als RDF/XML, Turtle, N-Triples, SPARQL, Content Negotiation
 - Stark konfigurierbar, anpass- und erweiterbar
 - Standardkonfiguration mit ausgeliefert
- Test it yourself:
<http://demo.dspace.org/data/handle/10673/5/ttl>
<http://demo.dspace.org/data/handle/10673/5/ttl?text>
`wget -O - --header='Accept: text/turtle' http://demo.dspace.org/jspui/handle/10673/5`
or download and install a release candidate

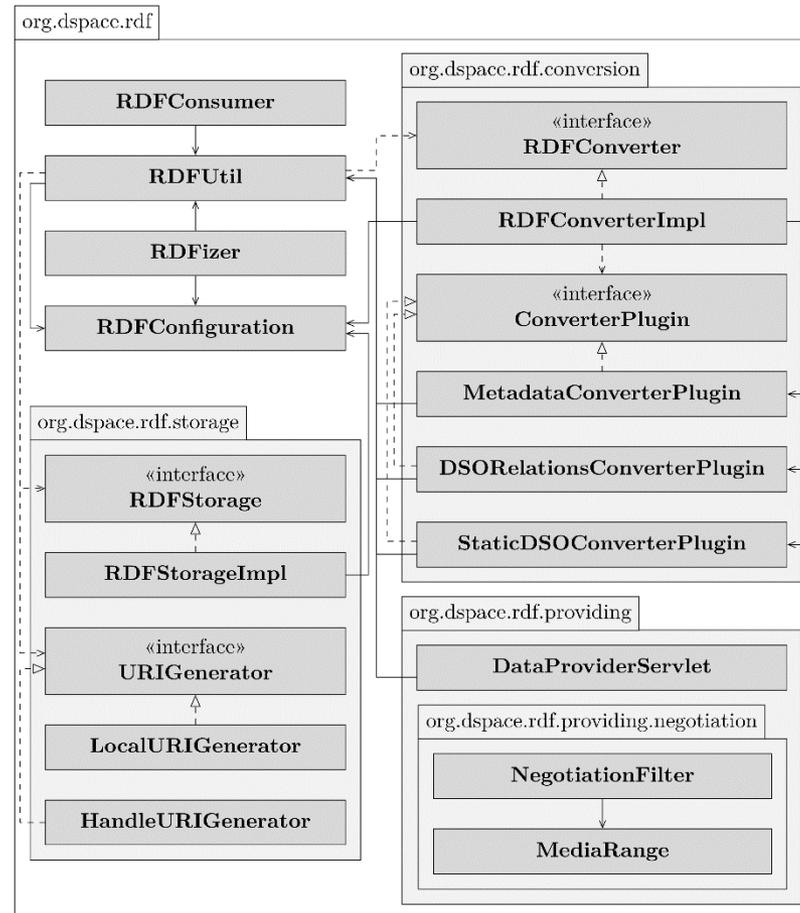
Wenn Sie DSpace 5.0 einsetzen, bitte schalten Sie den Linked Data Support ein!

Komponenten

- Daten werden bei Erzeugung und Änderungen automatisch in LD konvertiert
- Konvertierung erweiter- und stark konfigurierbar
- Verschiedene Module zur Erzeugung von URIs nutzbar (z.B. basierend auf DOIs, Handles oder den von DSpace erzeugten URIs).
- Triple Store als Cache ergänzend zu Datenbank
- `[dspace-install]/bin/dspace rdfizer` zur manuellen Steuerung
- Neue Webapp: `dspace-rdf` stellt Daten als RDF-Serialisate bereit
- Filter für Content Negotiation hinzugefügt (XMLUI und JSPUI)

- Noch fehlend: Link in HTML-Version (JSPUI/XMLUI) auf LD

Interner Aufbau (Klassenstruktur)



DSpace-RDF: Konfiguration (config/modules/rdf.cfg)

Konfiguration teilt sich auf:

- Verbindungsdaten zum Triple Store
 - Wie verbindet sich DSpace?
 - Wo befindet sich der öffentliche SPARQL Endpoint?
- Ein- und Ausschalten der Content Negotiation
- Wie sollen URIs zur Identifikation der Ressourcen in RDF generiert werden?
- Welche „Plugins“ sollen zur Konvertierung verwandt werden?
- Gegebenenfalls Konfiguration der Plugins

Verbindungsdaten

- DSpace nutzt für Abfragen an den Triple Store die SPARQL 1.1 Query Language und das SPARQL 1.1 Graph Store HTTP Protocol
- Dspace nutzt zur Manipulation des Triple Stores das SPARQL 1.1 Graph Store HTTP Protocol
- In der Datei rdf.cfg sind die Adressen für die entsprechende Endpunkte und ggf. Daten zur Authentifikation anzugeben
- Die Adresse des SPARQL Endpoints, wie er in RDF (z.B. mittels void:sparqlEndpoint) angegeben werden soll, sind im Rahmen der Konfiguration der Konvertierung anzugeben

Generierung von URIs

- Es können URIs des Repositoriums oder Persistent Identifier in Form von HTTP-URIs verwendet werden
- Dazu ist die Java-Klasse zur Generierung der URIs anzugeben:
 - *LocalURIGenerator*: Nutzung der „lokalen“ URIs des Repositoriums
z.B. <https://depositonce.tu-berlin.de/handle/11303/157>
 - *DOIHandleURIGenerator* / *DOIURIGenerator*:
Nutzung von DOIs (Wahlweise mit Fallback auf URIs oder Handles)
z.B. <http://dx.doi.org/10.14279/depositonce-1>
 - *HandleURIGenerator*: Nutzung von Handles (mit Fallback auf URIs)
z.B. <http://hdl.handle.net/11303/15>

Konfiguration der Konvertierung

- Konvertierung nutzt Plugins
- Alle zu nutzenden Plugins sind anzugeben, z.B.:

```
converter.plugins = org.dspace.rdf.conversion.StaticDSOConverterPlugin, \  
                    org.dspace.rdf.conversion.MetadataConverterPlugin, \  
                    org.dspace.rdf.conversion.SimpleDSORelationsConverterPlugin
```

StaticDSOConverterPlugin

Das StaticDSOConverterPlugin: ergänzt statische Daten (z.B. Angabe wo der SPARQL-Endpoint zu finden ist oder die Homepage des Repositoriums)

- Es können Daten angegeben werden, die zu allen Ressourcen hinzugefügt werden sollen
- Es können Daten abhängig vom Typ (Item, Collection, Community, Site) angegeben werden
- rdf.cfg enthält den Pfad zu Dateien mit den jeweiligen statischen Inhalten
- Dateien können in einer der gängigen RDF-Serialisierungen angegeben werden (per default Turtle)

SimpleDSORelationsConverterPlugin

Das SimpleDSORelationsConverterPlugin bildet Beziehungen zwischen DSpace-Objekten ab:

- Welche Top-Level-Communities hat das Repositorium?
- Welche Subcommunities und Collections gehören zu einer Community?
- Welche Items gehören zu einer Collection und welche Bitstreams zu welchen Items?
- Auch die andere Richtung wird abgebildet: zu welchen Collections gehört ein Item?
Zu welchen Communities eine Subcommunity oder Collection?
- ...

Das Plugin ermöglicht es die Linked-Data-Daten des Repositoriums durch das Verfolgen von Links zu erfassen, sich also in den konvertierten Daten von Resource zu Resource zu hangeln.

MetadataConverterPlugin

- Konvertiert die Metadaten von Items
- Kann zur Konvertierung von Bitstreams/Dateien erweitert werden
- Konfiguration wird in RDF angegeben, so dass der volle Umfang von RDF genutzt werden kann
- Konfiguration nutzt das DSpace Metadata RDF Mapping Vocabulary

DSpace Metadata RDF Mapping Vocabulary

<http://digital-repositories.org/ontologies/dspace-metadata-mapping/>

- Ein “Mapping” beschreibt die Konvertierung eines Metadatenfeldes in RDF
- Metadatenfelder können anhand Ihres Namens (key) und eines Regulären Ausdrucks erkannt werden, der auf den Wert des Feldes angewandt wird
- Erzeugt ein oder mehrere Triple
- Platzhalter für die URI des zu konfigurierenden Objekts
- Sowohl Literale als auch Ressourcen erzeugbar
- Literale können typisiert und Language Tags verwendet werden
- Die Sprachangabe zu Metadatenfeldern aus DSpace kann genutzt werden
- Der Wert eines Metadatenfeldes kann übernommen werden
- Reguläre Ausdrücke können zur Manipulation von Metadatenwerten bei der Erzeugung von Literalen und Resource URIs verwandt werden

Beispiel: DSpace Metadata RDF Mapping Vocabulary

@prefix dc: <http://purl.org/dc/elements/1.1/> .

@prefix dm: <http://digital-repositories.org/ontologies/dspace-metadata-mapping/0.2.0#> .

@prefix : <#> .

:title

dm:metadataName "dc.title" ;

dm:creates [

dm:subject dm:DSpaceObjectIRI ;

dm:predicate dcterms:title ;

dm:object dm:DSpaceValue ;

];

.

Beispiel: DSpace Metadata RDF Mapping Vocabulary

```
:doi
  dm:metadataName „dc.identifier.doi“ ;
  dm:condition „^doi:“ ;
  dm:creates [
    dm:subject dm:DSpaceObjectIRI ;
    dm:predicate dc:identifier;
    dm:object [
      a dm:ResourceGenerator ;
      dm:modifier [
        dm:matcher „^doi:(.*)$“ ;
        dm:replacement „http://dx.doi.org/$1“ ;
      ] ;
      dm:pattern „$DSpaceValue“ ;
    ] ;
  ] ;
```

Questions & Answers

Live-Demonstration

Technische Universität Berlin
Universitätsbibliothek
Pascal-Nicolas Becker
p.becker@tu-berlin.de

Servicezentrum Forschungsdaten und –publikationen
<http://www.szf.tu-berlin.de>

Repositorium DepositOnce
<http://depositonce.tu-berlin.de>

Diplomarbeit „Repositorien und das Semantic Web“
<http://www.pnjb.de/uni/diplomarbeit/>