Modeling gamification rules with ontologies for achieving a generic platform architecture

Patrick Mennig
Fakultät Informatik und Wirtschaftsinformatik
Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt

Gamification is an emerging trend for both, consumer and enterprise applications. The latter feature additional complexity due to their often highly distributed and interlaced nature. Herzig et. al. addressed this problem and proposed a generic platform for enterprise gamification. Based on their work, the problem of formulating gamification rules is taken up in this paper. Ontologies and semantic technologies are proposed as a possible solution. An ontology for gamification rules is created. Its applicability to solve the problem is validated with a thought experiment.

Categories and Subject Descriptors: Ontologies, Predicate Logic and Semantic Networks

Additional Key Words and Phrases: Gamification, Gamification Rules

1 GAMIFICATION IN ENTERPRISES

Playing is a popular human activity. The term ‘Play’ can be defined in various ways. An interesting aspect is the gratuitousness and the feeling of joy and fun [Kark 2011]. Not alone based on the roots in human history, applying game aspects and mechanics to – in the first place – atypical situations is an emerging trend. The rise began around 2010, mostly coined under the term gamification. Other variants are for instance “productivity games”, “applied gaming” or “playful design” [Deterding et al. 2011]. In its 2011 hype cycle, the technology advisory company Gartner introduced gamification as a technology trigger [Pettey and Goasduff 2011]. Despite gamification being a new trend in HCI, one will find several practical implementations. Among the popular examples is ‘Foursquare’, a platform to interact with friends and strangers based on checkins to various places [Anon. 2014]. One aspect of which is said to have led to its success (50 million users worldwide by May 2014) is the immersive use of gamification in its application. One may earn points, badges and other rewards for using the application and testing the boundaries of what is possible, finding secret rewards [Zichermann and Cunningham 2011]. Not only consumer applications benefit from gamification, but enterprises seek the integration of these concepts to increase employee engagement and productivity. The enterprise segment in the gamification market is projected to exceed the consumer segment by the end of 2013 [Peterson 2012]. Newer data could not be accessed due to high prices.

Author’s address:
Patrick Mennig, University of Applied Science Würzburg-Schweinfurt, Germany.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than FHWS must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission.
Gamification is often referred to as the “use of video game elements in non-gaming systems to improve user experience (UX) and user engagement” or the “use of game design elements in non-gaming contexts” [Deterding et al. 2011]. Another definition delineates it as the “process of game-thinking and game mechanics to engage users and solve problems” [Zichermann and Cunningham 2011]. The common denominator in all is the term ‘game’ or variants of it. Caillois distinguishes between two types of play activities: “paidia” and “ludus”, the former being more expressive and free form (referring to ‘play’) whilst the latter is bound to rules and a competitive rivalry (referring to ‘game’) [Caillois 1961]. Mapping this to the available definitions of gamification shows that ‘game’ other than ‘play’ was chosen correctly, as any gamification implementation has to include certain rules.

Seeking to define a generic platform architecture for enterprise gamification, Herzig et al. introduce a dualism between business and game rules. They demand an event engine, capable of describing work artifacts in a way, that any relevant action throws events $E = (u, x, p)$ with $u$ being the user doing the action and $x$ the step in a given process $p$. The events are routed to the business rule engine, which stores game rules $R = (r_1, ..., r_n)$ and processes received events $E$ by choosing which rule $r_i \in R$ should be applied. In their work, each rule is defined as $P \Rightarrow C$, with $P$ being the premise and $C$ the consequence to be executed (e.g. rewarding a certain amounts of points to the triggering user $u$). They further demand the rules to be “creatable and changeable for business users without technical background” allowing dynamical changes at runtime. Herzig and Schill propose a domain specific language, based on a taxonomy or ontology of game rules and mechanics [Herzig et al. 2012].

An ontology itself is a formalism that allows finding knowledge in data, construe it and enables a computer system to act based on semantic information. Taking the unavailability of such generic gamification platform and the coherence of ontologies being able to act based on generic semantics, leads to hypothesis H1 by abduction.

$H1$. A generic gamification architecture can be implemented with the usage of semantic technologies and ontologies.

In this paper, one valid ontology for concepts of ‘game’ relevant for enterprise gamification is crafted. An approach to define game rules as instances of ontology concepts is shown. It takes into account the propositional logic of the ontology itself to express gamification rules. The usage of an ontology as a base also helps to reduce overhead for creation of partial similar rules. To allow evaluation, requirements towards such an implementation, extracted from literature analysis, are shown and lead to a thought experiment, inductively validating the approach. Due to the restricted time available, the study abstains from a full deductive approach and limits itself to the inductive thought experiment.

2 CREATION OF AN ONTOLOGY FOR GAMIFICATION

Ontologies seek to define unambiguous representations of relations, symbols, real world objects, concepts (or classes) and their characteristics. The goal of those is to allow processing of information and data to understand its meaning and inherent semantics to generate knowledge. Allowing computer systems to understand data rather than parsing it following strict schemas, might finally lead to intelligent behavior. It is still required to add metadata to information, describing the context and semantics of given information [Dengel and Bernardi 2012]. Formalizing knowledge is called knowledge representation. Instructions for conclusions are
called inference rules. Facts describe the domain and represent the inherent knowledge. See listing 1 for three example rules. They are written as ‘subject-predicate-object’ sentences with, for instance ‘player’ being the subject, ‘actor’ the object and ‘is an’ the predicate. All rules follow this formalized syntax, in other words a formal specification.

The relations (predicates) in (A) and (C) are for humans obviously transitive, leading to the conclusion (E). That is the derivation of yet unknown from established knowledge and itself a fact. With that said, inference is fundamental and formal reasoning. The formality required to use it in computer systems is given by explicitly expressing the transitive nature of such relations. The rule (F) defines this and allows inference mechanisms in programs [Dengel et al. 2012].

Representations contain assertions about a given domain. In order to do this, facts, in form of objects and relations, and rules are defined. Inference mechanisms allow to conclude further facts. Mathematical logic is used to abstract mechanical knowledge gaining from human thinking [Dengel et al. 2012]. Consider the facts in listing 1 for an example: Let Jon be a sales agent, creating a new lead for his company. With just enough metadata a computer system could be able to conclude that he is participating in gamification while the actions of Jane might not be relevant for point calculation.

To give computer systems access to knowledge, it has to be projected from reality to a formalized semantic representation. The selection of concepts and properties modeled is defined by the goal of the system itself. Hence the resulting semantic model is less complex than the original domain through inherent abstraction. Still the structure and semantic of the resulting representation is to the greatest extent similar to the observed domain section [Dengel et al. 2012]. An ontology is the explicit specification of such domain concepts. The following findings F1 and F2 summarizes above remarks.

F1. Knowledge representation as an ontology allows to formalize domain concepts in a way that it is understandable for humans and computers.
F2. The formalized representation allows to conclude knowledge and facts from the given specification and (meta-)data.

The techniques to visualize and note ontologies vary. An ontology can be displayed as a semantic net, which itself is a graph \( G = (V, E, I, L, s, t) \). \( V \neq \emptyset \) is the set of objects \( O = (o_1, \ldots, o_n) \) and concepts \( C = (c_1, \ldots, c_m) \). \( E \neq \emptyset \) is the set of relations between objects \( O \) and/or concepts \( C \). The function \( l : V \cup E \rightarrow L \) assigns objects and/or concepts a name out of \( L \neq \emptyset \). Further \( s : E \rightarrow V \) and \( t : E \rightarrow V \) are functions referencing the start and end vertex \( V \) of an edge \( E \). Considering an edge \( e \), \( t(e) \) is the successor of \( s(e) \). A directed path \( P \) within the graph \( G \) is the sequence \( e_1, \ldots, e_n \) so \( s(e_i) = n_i \) and \( t(e_i) = n_{i+1} \) for \( i \in \{1, \ldots, n - 1\} \) [Stuckenschmidt 2009]. See figure 1 for an example of an ontology represented as a graph. This technique is human-readable and allows a quick reference and changes. The threshold for domain users to accept and work with graphs in a visual way is low [Rostanin and Weber 2012]. It is expected that domain experts and business users with basic understanding of formalisms used are able to create and maintain such ontologies, with regards to their domain understanding. This leads to the following finding F3.

F3. Knowledge representation as an ontology allows non-technical users to create and maintain computer-understandable specifications.

The same ontology can be expressed in various description languages like OWL, RDFSchema, F-Logic or SHIQ. Some of them are more restrictive than others, but the common point is the possibility to formalize ontologies and make them queryable [Baader et al. 2004]. They require a specific amount of knowledge about the technique itself which might not correlate with business users level of skills to stay maintainable. The tools used to express and visualize concepts and relations might affect this threshold positively. As the tool support depends on the individual decisions made in enterprises, this aspect is not further discussed here.

A straight way to translate semantic nets into logic formulas is based on the direct correspondence between graphs and relations, which itself can be expressed as a predicate. Each edge (i.e. an instance of a relationship) is understood as an atomic formula with two terms. The relationship ‘is-a’ would be transformed into the formula \( \text{is–a}(x, y) \) which results in \( true \) or \( false \), depending on \( x \) and \( y \). The concatenation of all atomic formulas in a given graph is done via the conjunction operator \( \land \). The full example in figure 1 would result in the following formula (1).

\[
\begin{align*}
is–a(\text{employee}, \text{player}) \land & \text{is–a}(\text{rival, player}) \\
& \land \text{is–a}(\text{player, actor}) \land \text{is–a}(\text{team, actor}) \land \\
& \text{is–a}(\text{actor, member}) \land \text{part–of}(\text{player, team})
\end{align*}
\]

(1)

Transforming this concrete formula into an universally valid deduction rule is done by replacing concrete instances with variables and adding operators for ‘exists’ \( \exists \) and ‘for all’ \( \forall \) increases the value of those. (2) and (3) are rules taken from the example in figure 1. (2) describes the transitive rule that each employee is a player and each player is an actor, hence each employee is an actor. The second rule describes the part-of relationship between a player and a team which again leads to transitive behavior.

\[
\begin{align*}
\forall x, y, z \text{is–a}(x, y) \land & \text{is–a}(y, z) \rightarrow \text{is–a}(x, z) \tag{2} \\
\forall x, y, z \text{part–of}(x, y) \land & \text{is–a}(y, z) \rightarrow \text{is–a}(x, z) \tag{3}
\end{align*}
\]
A shorter form to show transitive inclusion in classes is the implication by using concept names as unary predicates on concrete instances expressed by variables. The path $P = (e_1, e_2)$ with $t(e_2)$ being transitively dependent of $s(e_1)$ can be found in figure 1 as following: $\forall x \text{Employee}(x) \rightarrow \text{Player}(x) \land \forall x \text{Actor}(x) \land \forall x \text{Team}(x) \rightarrow \text{Actor}(x)$. Characteristic, i.e. necessitative properties of concepts are displayed as $\forall x \phi(x)$ which reads as ‘for all instances of a concept C applies the property phi’. Sufficient properties are expressed as $\forall x \phi(x) \rightarrow \phi(x)$, reading as for all instances with the property phi applies that they belong to the concept C. Necessitative and sufficient properties are expressed as $\forall x \phi(x) \leftrightarrow \phi(x)$. Cardinality can be expressed by combining the exist operator with binary comparison, for example: $\forall x \text{Team}(x) \rightarrow \exists y \text{part–of}(x, y), a = 2$ describes that a team consists of at least two members for whom applies the part-of relation [Stuckenschmidt 2009].

Using the possibilities of predicate logic to further formalize ontologies, it is possible to create powerful formulas. For a given instance $i$ of a concept $C$, one can decide whether transitive relationships or properties apply. By given properties, one can decide whether an instance $i$ is part of a concept $C$ or not. Hence, by formulating an ontology and parsing it into description logic it is possible to process it with computer programs. This leads to finding F4 and F5.

**F4.** Ontologies visualized as semantic nets (graphs) can be parsed into formulas. **F5.** These formulas can be used to decide for membership of classes and check necessitative/sufficient properties.

The findings F1-F5 lead to the suggested approach to implement a gamification rule engine, by creating a basic ontology for game and gamification concepts. To assert this approach, two further hypotheses are constructed:

**H1.1.** The ontology will be understandable for both, business users and information systems as well. Any adjustments that need to be made to an existing rule system will be done in a graphical environment, thus reducing the inhibition threshold of users entrusted with maintaining gamification rules. **H1.2.** Transitive rule definitions can be avoided by inheritance mechanisms of ontologies, e.g. ‘salesman is a actor’ and ‘Jane Doe is a salesman’) ‘Jane Doe is a actor’. Hence all rules for actors apply to Jane.

Stuckenschmidt proposes a process for creating an ontology. He differentiates between the conceptual level of an ontology and its formal level. The first targets the structure and terminology, while the latter focuses on the formal depiction in a valid description language. No coherent procedure model has been established yet, but his proposed process is an iterative approach of several steps that allow cyclic refinement [Stuckenschmidt 2009]. At first, the to be modeled section of the domain has to be defined in order to restrict the boundaries. Taking a closer look at what shall be modeled shows ‘game’, ‘gamification’ and ‘enterprise’ as the root domains. It is unnecessary to model all aspects of these, keeping in mind that certain specific aspects could be added once the need arises. Approaching what is not to be modeled helps reducing domain complexity. Bista et al. propose a gamification model $G$ as follows: $G := M, A, C, P, B, R_P, R_B >$ while $M$ is a member of a given community, $A$ is the set of possible actions one can perform in a context $C$. He earns points $P$ and badges $B$ by rules $R_P$ and $R_B$ [Bista et al. 2012]. The rules $R_P$ and $R_B$ will be implicitly and explicitly contained in the ontology itself and serve – as well as all other components of the seven tuple – as the constraints of domain concepts. The ontology will focus on members, their possible actions in respective contexts and the possibility to earn badges. Other concepts like
challenges or quests should not be considered in order to reduce complexity [Zichermann and Cunningham 2011].

Following the proposed process requires setting up competency questions. These are exemplary but still constructivistically for the resulting ontology. Defining beforehand what knowledge is desired to be contained. To create the competency questions domain experts are interviewed. This is done in form of an online questionnaire published on www.boardgamegeek.com, a community for board game enthusiasts. To focus on participants with high expertise in the field of games and game rules, their experience is recorded and only those with at least three years of regular game consumption are taken into consideration.

One objective of the expert interviews is to gather competency questions, describing beforehand what the ontology should be able to answer afterwards. Additional questions are found through literature research. The following list provides an overview of identified ones.

Q1. What is/are the goal(s) of the game?
Q2. Is a given X a player?
Q3. What are the consequences of action Y from player X.
Q4. Is player X part of a team?
Q5. Which rank has player X on a leaderboard Z?

Ensuing from these interviews and the study of literature, relevant terms of the domain are identified. Besides publications covering gamification ([Herzig et al. 2012], [Bista et al. 2012], [Zichermann and Cunningham 2011]), games ([Caillois 1961]) and game design ([Schell 2008]) rulebooks of board games are taken as research sources. The major findings are listed together with information about the sources and the frequency, i.e. how often they are mentioned across different sources. These are then sorted into concepts and relationships by figuring out which represent sets of objects and which not. This methodology lacks formalism, however it is best practice for setting up ontologies [Stuckenschmidt 2009].

After setting up the list of relevant terms and sorting out concept candidates, the class hierarchy is built. By asking ‘all X are Y; subsumption between concepts has to be noticed. This is constructivistically motivated and the result cannot be understood as ‘the’ class hierarchy rather ‘one possible’. This should not be considered as a drawback since this accounts for the flexibility in using ontologies to model concepts and relationships. Figure 2 shows the identified hierarchies of concepts. It does not yet include any other relationship than ‘is a’. Hence it only accounts for the understanding of subsumption between concepts. The complete hierarchy
is shown in figure 2 and its full discussion would leave the scope of this paper. An interesting aspect that has to be mentioned is that all concepts except for 'game' are subconcepts of 'mechanic.' The targeted ontology shall reflect games against the background of gamification in enterprise contexts. ‘Mechanics’ as the parts and pieces of any gamification implementation split into 'rewards,' 'goals,' 'operations,' 'rules,' 'leaderboards' and 'actors.' Reconsidering the defined boundaries $G = \langle M, A, C, P, B, R_p, R_g \rangle$ [Bista et al. 2012] shows high conformity.

Identified terms that did not match well into concepts are considered candidates for relationships. Their set is not limited to the list of terms as characteristic attributes are considered. With respect to the class hierarchy, this targets common properties of all members of a concept as well as attributes unique to members of a concept or unique when compared to other concepts. From the point of view of philosophy, these properties might be intrinsic or extrinsic. The first are constructivistically for a given object, losing such attributes means losing the very own character. The latter are those properties that are given to an object, but might change over time, without affecting the objects character [Stuckenschmidt 2009]. Both types are contained in the final ontology but used equitable. The closeness to IT-systems allows to consider extrinsic attributes being 'nullable' while intrinsic properties force a value other than 'null.' Relationships are attached as high as possible in the concept hierarchy to account for inheritance and minimize repetition.

Once the relationships between concepts are taken into consideration, the ontology’s complexity increases. The terms identified as relationships but yet unassigned between concepts are added. Defining beforehand which are applicable for certain concepts limits the possibilities. Figure 3 shows the ontology for gamification. One may notice certain aspects left out, like a property ‘name’ for ‘player.’ This is due to the fact that the ontology describes concepts, not instances and the ‘name’ of a ‘player’ $X$ as concept is irrelevant. Concrete implementations might add certain aspects typical for IT systems. Checking back with the competency questions
proves as an assertion of the created result. Considering Q1, the answer could be 'the goals are mechanics of the game' which is correct but unrewarding. Adding concrete instances of goals as well as quantification operators will result in a true assertion of (4). Q2 can be proved answerable the same way. Checking the class hierarchy and the existence of intrinsic properties allows to decide whether a given v is a player or not. The identical approach leads to the conclusion that the ontology created, can answer the competency questions Q1-Q5 and proves usefulness for the desired goals.

\[
\forall \nu, x, y, z \text{\ Goal}(\nu) \rightarrow \text{Mechanic}(\nu)
\]

\[
\exists \nu ' x \text{\ Goal}(\nu) \ \text{change}(\nu, \text{State}(x))
\]

\[
\exists \nu ' y \text{\ Goal}(\nu) \ \text{earn}(\nu, \text{Experience}(y))
\]

\[
\exists \nu ' z \text{\ Goal}(\nu) \ \text{increase}(\nu, \text{Points}(z))
\]

The ontology is still to be translated into predicate logic for formal reasoning. To take this into account, quantification operators as in (4) have to be added. Ontology engineering environments like Protégé assist the user to a great extent [Mizoguchi 2004]. Custom toolsets integrated in the domain software business users are familiar with, are imaginable. The targeted induction, i.e. the thought experiment following does not require to have these operators present for all relationships. Hence it is refrained from adding them to the whole ontology.

Summarizing previous section, ontologies have been examined for aspects of usability and convenience to fulfill requirements. F1-F5 led to the hypotheses H1.1 and H1.2. In order to approve or refuse them, an ontology for gamification has been created. The following section will investigate gamification implementation requirements and construct one concrete case for a thought experiment. This will indicate towards validation of H1.1 and H1.2.

3 GAMIFICATION THOUGHT EXPERIMENT
The posited approach of using gamification rules based on semantics in enterprise software and ontologies shall be evaluated for one specific case with a thought experiment. Referring to the outset, the proposed architecture of Herzig et. al. is taken into consideration. They demand an event system with a message broker and a separate gamification rule engine reacting once receiving events. The goal is lose coupling of components and instant feedback [Herzig et al. 2012]. Active broadcasting of events is mainly based on the demand for decoupling single system elements and shall be replaced by adding semantic information to system logs, database tables and existing events. The requirements towards gamification highly vary, depending on the source used, from points, redeemable points, social points, badges, leaderboards, player interaction, level, progress, challenge, reward, instant feedback, story [Scheiner et al. 2012] [Zichermann and Cunningham 2011] [Hamari et al. 2014] to virtual goods, teams, time pressure, gift and marketplaces [Herzig et al. 2012]. Due to the limited boundaries of the ontology fixed to \( G = \langle M, A, C, P, B, R_p, R_b \rangle \) [Bista et al. 2012] the following case is constructed:

Observed is a company Co that has \( n \) employees \( E_{Co} = \{E_1, ..., E_n\} \) working with an enterprise software system \( S_{Co} \). They perform actions \( A = \langle S_{Co}, E, C \rangle \) within \( S_{Co} \). The context \( C \) describes the source, the target and temporal aspects of \( A \) and is added together with semantic information to a specific log file \( L_{S_{Co}} \) that is accessible by \( S_{Co} \) and the gamification rule engine \( G \). It has a set of rules \( R_{Co} = \langle R_p, R_b \rangle \) defined as instances of the concept 'rule', a leaderboard (instance of concept 'leaderboard') \( L_{bCo} \) and a list of \( m \) badges (instance of concept 'badge') \( B_{Co} = \{B_1, ..., B_m\} \) each employee has earned.
Assume Jane Doe is the employee $E_1$ at Co. She administrates $G$ and creates a new instance $R_{P1}$ of 'rule' covering $A_1$. $E_1$ utilizes a graphical environment that shows the existing ontology's concepts. $E_1$ is able to add the new instance via drag-and-drop or a wizard. Once $R_{P1}$ is created, Jon Doe, $E_2$, starts working in $S_{Co}$. He is working in sales and has a new potential customer, hence he generates a lead ($A_1$) in $S_{Co}$ which outputs a new entry containing $C$ in $L_{Co}$. $G$ detects the change in $L_{Co}$ and parses the entry. Due to semantic information, $G$ finds the following knowledge: $E_2$ is an employee, hence he is an actor, hence he has an amount of points $P_{E2} = 0$; $R_{P1}$ allows $A_1$; $A_1$ is an operation which increases points; $C$ determines that $E_2$ has ended $A_1$ in less than 5 minutes. Based on this knowledge $G$ decides to increase $P_{E2} = 10$. The change in $P_{E2}$ defines $E_2$ in $L_{b_{G1}}$ and allows him to advance one rank, overtaking another colleague.

Above thought experiment inductively shows for one concrete case the applicability of semantics and ontologies for gamification rule engines. Therefore H1.1 cannot be rejected – at least in this case it has to be considered as valid. For a general acceptance, a correct deduction has to be done. This paper also shows that H1.2 can be accepted in this case. Inheritance mechanisms used in the semantic net (‘is-a’) allow automatic parsing of transitive relationships. As H1.2 was mainly accepted due to findings in literature, it is suggested to be accepted generally until falsified. This case also allows to accept H1, as within the defined boundaries, the suggested approach was able to fulfill the requirements. To further support H1, one should do an implementation and perform field experiments.

4 PROSPECTS ON A GENERIC GAMIFICATION RULE PLATFORM

As discussed in the beginning, gamification is an emerging trend, increasing the need for software support. Reducing coupling between enterprise systems points towards a generic gamification platform approach. This paper investigated one possibility towards a generic gamification rule platform by using ontologies and semantics to loosely couple the involved systems. The hypotheses crafted in section 1 and 2 could be accepted for now, based on an induction with a thought experiment. It is necessary to further investigate ontologies as an approach to generic platforms, in order to falsify or back up the proposed hypotheses.
REFERENCES

Anon. 2014. About foursquare.


