Technical Report: GuessWhat?!

Human Intelligence for Mining Linked Data

Thomas Markotschi and Johanna Völker

KR & KM Research Group
University of Mannheim, B6 26, 68159 Mannheim, Germany

Abstract. Ontologies are an important prerequisite for an increasing number of knowledge-intensive applications, not to mention the great vision of the Semantic Web. However, despite the obvious need of such formal and explicit representations of knowledge, many people refrain from investing into the tedious and time-consuming task of ontology engineering. At the same time, purely automatic means for ontology construction so far have failed to meet our expectations in terms of quality and expressivity. In this paper we describe *GuessWhat?!*, a multi-player online game in the tradition of *semantic games with a purpose*. By leveraging people's play instinct it motivates them to contribute to the creation of formal domain ontologies from Linked Open Data. We detail on the implementation of the game and present the results of an initial user study.

1 Introduction

In 2001, Tim Berners-Lee [1] introduced the term *semantic web* in order to refer to what is now perceived as the future of the internet: a web of machine-interpretable content that can be processed by automatic agents in a meaningful way. Since achieving this ambitious goal requires both an explication and formalization of relevant domain knowledge, ontology languages such as RDFS [3] and OWL [11] have emerged as a means for unambiguous knowledge specification. However, the realization of the semantic web as envisioned by Tim Berners-Lee and the wide-spread use of intelligent, reasoning-based applications is still hampered by the lack of ontological resources.

The vast amount of *linked data*[1] in the form of RDF triples which is out there on the internet can be considered an important step forward on the way to the semantic web. In fact, a huge number of mashups and applications already benefit from billions of triples in the repositories of DBpedia[2], Freebase[3] or the like. At the same time, several applications, especially in the complex domains of medicine or bioinformatics, demand for more formal and expressive knowledge representations which are highly accurate in terms of syntax and semantics – a crucial prerequisite for logical inference yielding non-obvious conclusions. Constructing such representations, i.e. ontologies, of sufficient quality, size and expressivity is a very challenging endeavor. Making high

[1] `http://linkeddata.org`
[2] `http://dbpedia.org`
[3] `http://www.freebase.com`

demands on scarce human resources and the expertise of ontology engineers it is extremely expensive and time-consuming. While sooner or later automatic approaches to ontology construction (*ontology learning* [5]) could help to overcome this knowledge acquisition bottleneck, these approaches so far have failed to meet the expectations of people who argue in favor of powerful knowledge-intensive applications.

Semi-automatic approaches leveraging human intelligence and the *wisdom of the crowds* seem a particularly promising way to increase the efficiency and effectiveness of knowledge acquisition. A pioneer in the field of crowdsourcing [12] was Luis von Ahn who suggested to exploit the play instinct of humans for computationally difficult tasks by so-called *games with a purpose* [19]. His ideas were later taken up by Siorpaes and Hepp [16] who created the first *semantic* games with purpose: multi-player online games as incentives for human participation in the acquisition of formal and explicit representations of knowledge.

In this paper, we present *GuessWhat?!*, a novel semantic game with a purpose which leverages both human intelligence and collaboratively created data for bootstrapping the semantic web. *GuessWhat?!* motivates people to contribute to the creation of a domain ontology: Presented with class expressions such as `fruit AND yellow AND grows on tree` automatically generated from Linked Open Data the players have to invent as quickly as possible a suitable class name (`banana` or `lemon`, for example). This can be quite challenging as the generated descriptions, which are fairly general in the beginning (e.g. `fruit`), become more and more specific as the game proceeds (e.g. `fruit AND yellow`). As soon as a player cannot think of a suitable label anymore, he or she has lost the round, and finally, the player who after multiple rounds, has come up with the highest number of plausible class labels wins the game. Note that the rules of this game are inspired by a well-known card game.[4] We modified them in order to enable the verification and labeling of automatically created class expressions by people who do not even need to be ontology experts. Initial user studies give raise to the hope that *GuessWhat?!* will make *semantic web mining* [17] a lot more fun in the future.

The remainder of this paper is structured as follows. Section 2 gives an overview of related work in the field of automatic and semi-automatic knowledge acquisition. In Section 3, we outline the rules and implementation of our semantic game with a purpose, *GuessWhat?!*. Section 4 describes the results of our evaluation experiments, and finally, we conclude with a summary and an outlook to future work (cf. Section 5).

2 Related Work

Aiming at the semi-automatic acquisition of terminological knowledge from linked data, we find our approach related to a considerable amount of work on *ontology learning* [5], i.e. the automatic or semi-automatic generation of ontologies by machine learning or natural language processing techniques. The vast majority of existing methods have been developed to facilitate the extraction of ontologies from unstructured text [4], but only few of them support the acquisition of logically complex class expressions [18]. This also holds for early attempts to generate ontologies from linked data, e.g.,

[4] *Ein solches Ding* by Urs Hostettler (1989)

by means of systematic generalization [7], clustering [13] of RDF data, or more recent work on selective ontology reuse [14].

Other logical approaches based on *Inductive Logic Programming* (ILP) [6, 8] combine machine learning and logic programming techniques in order to derive class expressions from positive and negative examples (e.g. individuals known to instantiate the target class). Although ILP-based methods have already been shown to yield good results when applied to linked data [9, 10], most implementations are inferior to statistical approaches in terms of scalability and robustness. Moreover, ILP is not per se an interactive approach – a fact that makes it very difficult for these techniques to handle incomplete or incorrect knowledge at runtime. An alternative to the automatic generation of class expressions are natural language interfaces allowing users to interact with an ontology editor by means of controlled natural language (e.g. [2]). The drawback of these approaches is that people have to invest into learning a syntactically and lexically restricted language. Therefore, strong incentives might still be required in order to motivate people to formalize knowledge.

One of the strongest incentives is money or any type of financial benefit, as witnessed by *crowdsourcing* applications such as *Amazon Mechanical Turk*.[5] This service provides programmers with the opportunity to create so-called *Human Intelligence Tasks* (HITs), i.e. tasks that are not yet solvable by purely computational means. Such HITs can be anything from choosing the best category for a specific product over validating addresses to a fun quiz about celebrities. Other applications use "cheaper" incentives like fun and entertainment to attract people. *Games with a purpose* first introduced by Luis von Ahn [19] have been invented in order to leverage the play instinct of humans for tasks such as image labeling, solving captchas or the tagging of audio or video files. The ideas of von Ahn were picked up by Siorpaes and Hepp [15] who pioneered the field of *semantic games with a purpose* by suggesting to turn ontology acquisition into a fun game. One of their games, *OntoPronto*, motivates people to link Wikipedia articles to concepts of an upper-level ontology, while another one has been invented to facilitate the annotation of YouTube videos with respect to their genre or language.[6] Even more games are currently being developed in the EU project *Insemtives*.[7]

3 GuessWhat?!

In the following we will elaborate on the design and implementation of *GuessWhat?!*, a novel game with a purpose that leverages human intelligence for mining linked data. After introducing the rules of the game (cf. Section 3.1), we will turn to the software architecture and describe in detail the algorithms underlying the computational intelligence of *GuessWhat?!* (see Section 3.2).

3.1 Rules of the Game

GuessWhat?! can be played with a minimum of at least two players and currently has no limitation on how many users are allowed to participate. Each gaming session consists

[5] `http://www.mturk.com`
[6] `http://www.ontogame.org`
[7] `http://www.insemtives.eu`

of one or more *rounds* in the course of which the players have to guess the name of an unknown concept partially described by the game engine. Note that in most cases there will not be *one* correct answer, but many possible solutions – namely all the concepts which match the given description. When a round has ended, the players evaluate each other's answers in terms of plausibility, before starting with a new round and a new concept description.[8] More specifically:

Guessing: At startup, the players are presented with a partial description of a concept like, for example, `tangible`, `animal` or `used for transporting people`. Now, each player is asked to think of a "fitting object", i.e. a concept which matches the description, and to enter its name into the user interface. Alternatively, a player may choose "pass" in order to indicate that he or she does not know what the description might refer to. When every player has given an answer, the initial description is extended in a way that it becomes more specific (e.g. `animal AND carnivore`), and again each participant in the game needs to come up with a plausible label for the class denoted by the description. A *round* ends when either every player passed on the same description (e.g. nobody can imagine something that is `animal AND carnivore AND NOT dangerous AND poisonous`) or a previously defined maximum description length has been reached.

Evaluation: In the subsequent evaluation phase the players are asked to judge the final answers of their opponents. In particular, a player has to decide for each concept name entered by an opponent whether or not it fits the class expression that has been generated by *GuessWhat?!* until the moment when the round ended. The possible choices are accept ("OK"), reject ("Not OK") or abstention ("I don't know"). If he or she decides to reject an answer, the evaluator has to specify which part of the class expression conflicts with the given answer (see Figure 1). After the evaluation phase, a new round with a fresh description begins. To hold up a certain game flow, the last player who has not finished his task (i.e. answering or evaluating) is faced with a ten second timeout. If he fails to beat the clock he automatically "passes" or chooses "I don't know" in the evaluation phase.

The development of *GuessWhat?!* was motivated by the lack of formal terminological knowledge on the semantic web. The players' answers during the various game rounds and subsequent evaluation phases give us the opportunity to not only obtain valuable feedback with respect to the meaningfulness of the generated class expressions (as we will see in Section 3.2, these are automatically generated from linked data), but they also enable us to link complex descriptions to atomic concepts in an ontology. Note that the expressivity of the class descriptions generated by *GuessWhat?!* is not limited to conjunctions. Imagine, for instance, that during one of the rounds, three definition fragments `tangible`, `fruit` and `yellow` have been presented to the participants of *GuessWhat?!* altogether forming the class expression `tangible AND fruit AND yellow`. Further let us assume that the final answers of the players are `banana`, `lemon` and `cherry`. Now, during the evaluation phase that follows, the first two of these answers could be accepted as both bananas and lemons match the proposed description. The last answer, `cherry`, should rather be rejected as it is

[8] In the remainder of this paper, we will occasionally use the OWL terminology and refer to these (semi-formal) descriptions of concepts as *class expressions*.

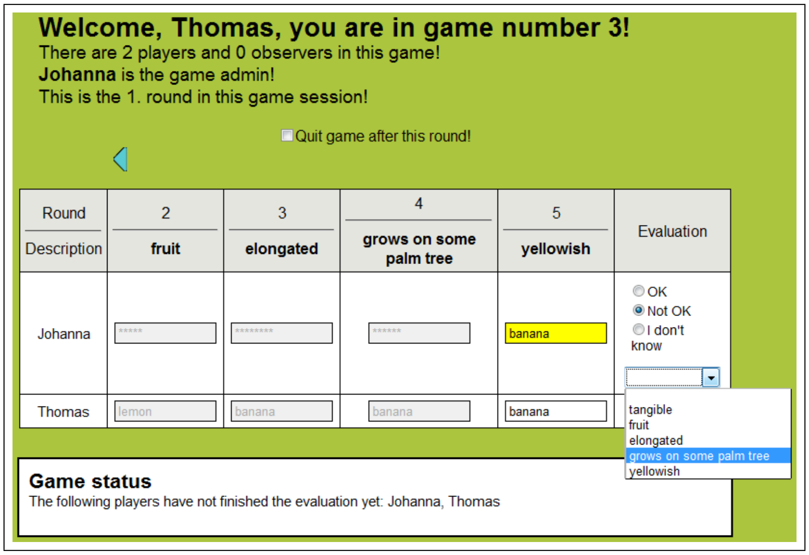

Fig. 1. A screenshot of *GuessWhat?!* taken during an evaluation phase.

not `yellow`. If one of the players notices this mismatch between something being both yellow and a cherry, and explains his judgement accordingly (i.e. by selecting one or more[9] parts of the description which contradict the other player's answer) we can not only conclude that `banana` and `lemon` are `tangible AND fruit AND yellow`, but also that `cherry` must belong to the class `tangible AND fruit AND NOT yellow`. Further screenshots as well as detailed instructions concerning the user interface of the game can be found online.[10]

3.2 Implementation

We developed and implemented the game in Java. Figure 2 shows the layered architecture of the game that runs on an Apache Tomcat 6.0 web server. The *data layer* consists of a Sesame RDF Store and a MySQL database. The connectors for these data stores can be found in the *data access layer* which also contains several components for gathering RDF triples from external semantic resources. The definition mining implementation in the *business logic layer* accesses the collected data, stores it in internal repositories and generates a class expression. The two beans which also belong to this layer are responsible for handling the user inputs which are made via the graphical user interface. In the following, the most essential components are discussed in more detail.

Data access and storage. For the creation of each class expressions we use a "seed concept" that serves as a starting point of the data gathering process.[11] This way we make

[9] The next version of the user interface will allow for multiple selection.
[10] `http://nitemaster.de/guesswhat/manual.html`
[11] In our experiments, these concepts were picked by hand but they could also be chosen randomly from a dictionary or an existing ontology.

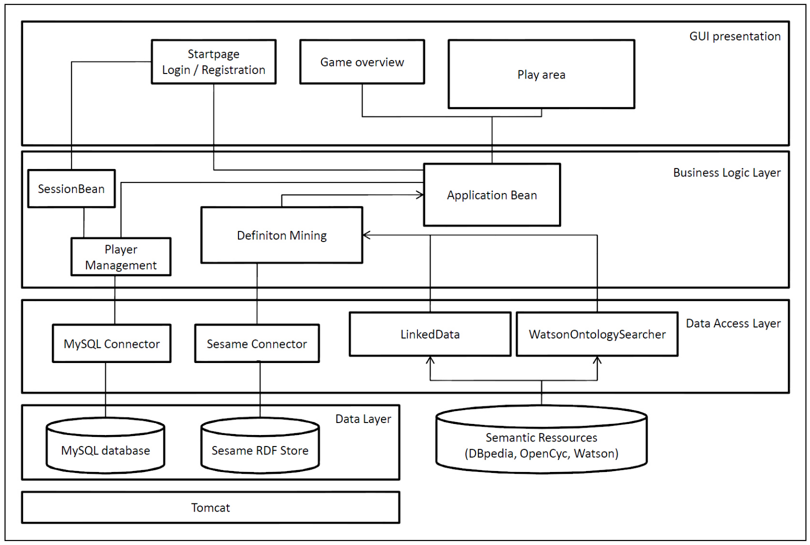

Fig. 2. The architecture of *GuessWhat?!*.

sure that the generated class expressions are mostly meaningful as otherwise people might be bored with a lot of nonsense descriptions. The collection of data from external resources such as *Linked Open Data* is mainly done via a local SPARQL endpoint[12] and consists of several steps:

1. For each of the seed concepts (e.g. `banana`), try to find a matching URI in *DBpedia*, *Freebase* and *OpenCyc*. On the one hand, this is achieved by querying the search engines of these data stores. On the other hand, further URIs can also be retrieved by following data links such as `owl:sameAs` relations in the results of these searches.
2. These initial URIs are afterwards queried with SPARQL queries to retrieve as much information as possible about the concept. Given a concept C_1, concept C_2 and property P_1 are searched for in the following order:
   - C_1 owl:equivalentClass C_2
   - C_1 rdfs:subClassOf C_2
   - P_1 rdfs:domain C_1
   - P_1 rdfs:range C_1
   - C_2 rdfs:subClassOf C_1

   At first, concepts which are equivalent to the original one are searched for through step 1. Afterwards all superclasses of the original concept and the ones that were just discovered are extracted. Properties whose domain or range is associated with a concept which has been found so far are searched for in step 3 and 4. At last, the subclasses to the gathered concepts are extracted. This sequence is maintained to ensure the maximum possible amount of extracted data. If for example step 2

[12] `http://www.openlinksw.com/virtuoso/`

and 4 would be switched, valuable information (e.g. properties of superclasses of the original concept) would not be found. The URIs of the concepts that have been gathered are directly added in a queue and are queried against afterwards for further information retrieval.
3. Store the gathered information in a repository for faster access.

Definition mining. This component is responsible for generating class expressions from the individual pieces of information collected by the data access component described further above. The procedure of assembling the various bits and pieces collected from the various sources into a coherent description requires several steps:

1. Analyze the labels and URIs of the classes and properties that were retrieved before by means of simple natural language processing. The purpose of this step is to identify expressions which can be translated into logical operators (e.g. negation or disjunction), as well as to break down complex descriptions (e.g. long class labels found in OpenCyc) into smaller fragments. This way, we can avoid redundancy when assembling the overall class expression and compute more meaningful statistics for ranking the various aspects of a concept's description.
2. Judge the smaller fragments with respect to *generality* and *confidence* (i.e. relevance as to the seed concept, see below). This information is required to ensure that the individual parts of a description become more specific as a round goes on and players are not presented with overly specific descriptions (e.g. `prepared from some tea leaves`) in the beginning or very high-level descriptions (e.g. `tangible`) at the end of a round.

For example, consider the superclass `an elongated yellowish fruit` which was found during the search for information about the seed concept `banana`. Using the LExO [18] approach, we can split the class label into `elongated`, `yellowish` and `fruit`. In order to compute the confidence and generality scores of these fragments, the extracted data is joined in one big tree structure. The graph mining algorithms applied to this structure take into account the following aspects:

– What is the distance of a class or property to the seed concept?
– How often was a class (e.g. `an elongated yellowish fruit`) or property found during the search for information about a concept?
– How often is every single fragment of its description (e.g. `elongated`, `yellowish`, `fruit`) present in the result set?
– How many paths from the seed concept to the root node (i.e. `owl:Thing`) does a class or property lie on?

The first item in the enumeration above is referred to as "generality". The $generality(c)$ of a concept c represents its importance in the graph with regard to its "distance" to the seed concept. The distance of a class or property to the seed concept is determined by taking the longest path from the class to the seed concept. Imagine the following simple taxonomy: The seed concept `banana` has two superclasses `fruit` and `shaped object`. Furthermore, the class `object` is a superclass to `shaped object`. Both `object` and `fruit` are direct subclasses to the root node owl:Thing. The distance of fruit and shaped object therefore is 1, the one of object is 2. By using Formula 1, where $maximumDistance$ is the highest value of any concept, we, for example, receive $generality(fruit) = 1 - (1/2) = 0.5$. The generality value ranges between 0 and 1 and the higher it is, the more specific is the respective fragment of the concept description.

$$generality(c) = 1 - \frac{maximumDistance(c)}{maximumDistance} \tag{1}$$

The factors 2 - 4 of the enumeration above together form the "confidence" score of a concept and define how good its label fits to the seed concept. Factor 2 (referred to as "simple score") is calculated by dividing the number of the concept identifier's occurrences, which were found during the data extraction, by the maximum number of occurrences of any identifier. The calculation of the third point (referred to as "complex score") is similar. However, for this score, the occurrences of the fragments which were created by the lexical transformation are counted. So if for example the label of the concept with the identifier Banana was Yellow Fruit and the fragments resulting from the transformation therefore were Yellow and Fruit, the occurrences of these would be counted in all transformed labels and summed up. This value is then divided by the maximum score of any concept. The last factor (referred to as "path score"), is calculated by dividing the amount of unique paths, a concept lies on from the seed concept to the root node (i.e.owl:Thing), by all unique paths which go from the seed concept to the root node. These three values range between 0 and 1 and form the confidence score of a concept by means of Formula 2.

$$confidence(c) = \frac{simpleScore(c) + complexScore(c) + pathScore(c)}{3} \tag{2}$$

From both confidence and generality we compute, for each fragment of a description, an overall score which changes as the game proceeds: Imagine for example the seed concept banana and the two fragments tangible and fruit. The confidence of tangible is rather high (as it was found quite frequently) while its generality score is comparatively low (i.e. it is not very specific). For fruit it is the opposite. Now, in the beginning of the round (*step* = 0), tangible is favored over fruit, that comes into play later when $step$ approaches $step_{max}$. This way of balancing confidence and generality is expressed by the following formula:

$$score(c, step) = (step_{max} - step) * confidence(c) + step * generality(c) \tag{3}$$

User interface. The user interface has been fully designed in XHTML and uses some components of the *ICEfaces*[13] framework. The latter also includes an AJAX Push implementation which is used to exchange data with the server in near real-time and to update the graphical user interface on the client-side. The execution of the game logic is handled by two independent types of beans. The application bean implements the Singleton pattern and is only initialized at its first access. It coordinates everything concerned with the game execution such as managing players, game creation and handling inputs. Additional session beans, which are initialized for every connected user, are responsible for the login and registration.

[13] http://www.icefaces.org

4 Evaluation

We will now summarize the user feedback which we gathered throughout the implementation and testing process (cf. Section 4.1), before taking a closer look at the results of these test sessions (see Section 4.2). The complete data set acquired during the evaluation of *GuessWhat?!*, including the automatically generated class expressions as well as the players' answers and the ontology constructed thereof is available online.[14]

4.1 Gaming Experience

In order to evaluate the gaming experience and the incentives created by *GuessWhat?!*, we scheduled several test sessions with different groups of people – ontology experts as well as users without any prior knowledge about semantic technologies.

First, two "beta tests" were conducted with 5 players participating in each of them. Afterwards, we asked all of the participants for their experiences throughout the game. They complained about the description fragments in the game being too complex and they told us that many of those did not make much sense, and we were surprised to see that the players of the first round were not enthusiastic about the game. However, as the players suggested several improvements to make the game more appealing, we learned a lot from their feedback and re-designed some parts of *GuessWhat?!* right after this first test session. In particular, the description extraction mechanism has been greatly improved to generate much more simple fragments which are presented to the players. Additionally, game components such as a timeout to prevent dead-locks or a chat function for communication has been added. When we had finished the implementation of the revised, second version of the game, we conducted two more test sessions, each of them with 6 participants. In order to help us evaluate the gaming experience, the players were asked to fill out the questionnaire presented in Table 1.

In total, 10 players filled out the questionnaire. The most striking findings of this survey are summarized below. While the number of answers was too small for generating meaningful statistics, the feedback was mostly positive:

- We found no correlation between the players' prior knowledge about ontologies and their understanding of the game rules.
- None of the players disliked the game concept per se.
- A few people found that the game got boring after a while, but most of them were willing to play again soon.
- The generated descriptions made sense to the users in most of the rounds.
- The majority of players found that the others judged their answers in a fair manner.

4.2 Acquired Knowledge

During the various test sessions which we conducted in order to evaluate the "fun factor" of the game, an overall number of 59 class expressions was generated and labeled by the players. Table 2 shows a subset of these descriptions along with their corresponding seed concepts as well as the labels guessed by the players. For example, given the

[14] `http://nitemaster.de/guesswhat/data.html`

1. What is your experience with ontologies?
   Well experienced / No expert / No knowledge about ontologies
2. Are the game idea and the rules comprehensible?
   Yes / Learned by doing / No
3. How many rounds did you play?
4. How many players participated in your game (including yourself)?
5. Did you enjoy playing the game?
   Yes / Only in the beginning / No
6. Would you like to play the game again?
   Yes / No
7. Do you think that the order of the definition fragments did
   make sense? (i.e. getting more and more specific over time)
   Yes / Sometimes yes, sometimes no / Mostly not
8. Did you find it hard to answer?
   Yes / Sometimes / No
9. Do you think the other players' evaluation was fair?
   ○ *Yes / Sometimes not / No*
10. Please point out problems that you experienced while
    playing. (e.g. technical problems)
11. Please point out what could be improved, especially
    if you did not enjoy playing the game.

Table 1. The questionnaire for evaluating the second version of *GuessWhat?!*.

description that was generated for the seed concept `photo`, one participant of the game thought of a *picture of water*, while the other players said *image*, *poster* or *map* respectively. All of these answers are plausible and thus can be used to extend the ontology. For example, given the description that was generated for the seed concept `horse`, one participant of the game thought of a *mule*, while the other players all said *horse*. Note that not every fragment of the class expression makes perfect sense from a formal point of view. Some of the errors were introduced by misleading class labels, the extraction of contradictory facts or by the false classification of words during the natural language processing. However, as the players are asked to find fitting answers, they are held to recognize such malformed expressions and react by passing and ending the round.

In some cases the concept names provided by the players seem to denote concrete individuals rather than classes (e.g. *Focus*, a German magazine). Ideally those should be recognized and handled appropriately. Several of the other concept names do not really match the original description, like *milky way*, for example, which is not a type of `plasma`). This fragment of the class expression generated for the seed concept `star` has been extracted from OpenCyc, according to which a star is a kind of plasma.[15] Finally, not every fragment of the class expressions suggested by *GuessWhat?!* makes perfect sense from a formal point of view. Several errors were apparently introduced when long class labels were split into their semantic constituents (e.g. `containing stories AND articles`). Despite the above mentioned problems, many of the generated descriptions can be represented by means of OWL in a relatively straightforward way. For example, the class expression `device AND solid`

[15] `http://sw.opencyc.org/concept/Mx4rvVi80ZwpEbGdrcN5Y29ycA`

`AND tangible AND user_guided AND (egg_shaped OR round)` which was assigned to the concept `ball` by the players could be formalized as follows:

```
<owl:Class rdf:about=”ball”>
 <rdfs:subClassOf rdf:resource=”device”/>
 <rdfs:subClassOf rdf:resource=”solid”/>
 <rdfs:subClassOf rdf:resource=”tangible”/>
 <rdfs:subClassOf rdf:resource=”user_guided”/>
 <rdfs:subClassOf>
  <owl:Class>
   <owl:unionOf rdf:parseType=”Collection”>
    <rdf:Description rdf:about=”egg_shaped”/>
    <rdf:Description rdf:about=”round”/>
   </unionOf>
  </owl:Class>
 </rdfs:subClassOf>
</owl:Class>
```

5 Conclusion

As noticed by Buitelaar and Cimiano [4], the implementation of appropriate user interaction paradigms is among the greatest challenges for today's ontology learning approaches – let it be ontology learning from text or from structured resources. This is partly because automatically approaches are still far from achieving the accuracy that humans have in any knowledge modeling task. Also, the realization of the semantic web vision is such an ambitious goal that it seems indispensable to involve more people than just a handful of professional knowledge engineers. Especially domain experts without any prior knowledge about formal semantics and ontology representation languages must be enabled to contribute to the construction of ontologies.

In this paper, we presented *GuessWhat?!*, a semantic game with a purpose which has been developed in order to facilitate the construction of ontologies by people without profound knowledge in the field of semantic technologies. By hiding the complex syntax of ontology representation languages under the surface of an entertaining multiplayer online game, it makes knowledge acquisition easier and a lot more fun. In our opinion, this way of combining the wisdom of the crowds with semantic web mining is a very promising paradigm for future knowledge acquisition. Initial user studies indicate that a game like *GuessWhat?!* can be a lot of fun and that it might even raise awareness for semantic technologies among people who have never thought about problems such as the knowledge acquisition bottleneck or the semantic web.

Still, many technical and conceptual enhancements are left for future work. For example, we plan to redesign the current scoring system in order to improve the longterm motivation of the game, and to reduce the temptation of cheating, e.g., by an unfair evaluation of the rivals' answers. This is quite important as the overall success of the game with respect to the purpose of knowledge acquisition crucially hinges on the reliability of the information that can be obtained during the game. Moreover, we would like to conduct another user study, as we hope that more data (i.e. collected from a lot more users or within a longer timeframe) will enable the investigation of new methods for

photo	`resource AND depiction AND source AND tangible` `AND solid AND spatially continuous AND graphic`
Players:	*picture of water*, *poster*, *map*, *image*

bed	`physical object AND intentionally made AND furniture`
	`AND object within room AND four legged flat frame AND mattress`
	`AND used for sleeping on AND NOT natural AND NOT animate`
	`AND used on everyday basis`

Players:	*bed*, *steel bed*, *nail bed*, *ferric bed with matress*, *cocaine*

cloths	`woven AND sheet of some substance`
	`AND medium amount of bio deterioration resistance AND spatial`
	`AND topic AND generic`

Players:	*nylon bedsheet*, *cloth*, *jack wolfskin jacket set*

star	`heavenly body AND any of luminous celestial object`
	`AND seen on some sky AND astronomical AND spatially bounded`
	`AND plasma`

Players:	*proxima centauri*, *milky way*, *plasma rocket disguised as angle*
kitchen	`area AND set off walls within building AND room` `AND food preparation AND (home OR restaurant) AND indoor location`
Players:	*kitchen*, *garden house room*
toilet	`tangible AND disposal AND apparatus AND consisting of bowl` `AND fitted AND hinged AND seat`
Players:	*full garbage can*, *single-use camera*, *trash can*

magazine	`periodical publications AND containing stories AND articles`
	`AND often published AND (monthly OR bimonthly) AND journal`
	`AND institution AND publisher`

Players:	*PM*, *Focus*, *Bravo*, *paper*, *comic*

Table 2. Examples of class expressions and the seed concepts they were generated from. The following rows, starting with “Players”, list the labels assigned by the players.

mining semantics from the players’ behavior (e.g. considering answer times). We are confident that such a bigger user study will also provide us with additional arguments for many of the conclusions we have drawn from our preliminary experiments.

Acknowledgements Johanna Völker is financed by a Margarete-von-Wrangell scholarship of the European Social Fund (ESF) and the Ministry of Science, Research and the Arts Baden-Württemberg.

References

1. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
2. Abraham Bernstein and Esther Kaufmann. GINO – a guided input natural language ontology editor. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *Proceedings of the 5th International Semantic Web Conference (ISWC)*, volume 4273 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2006.
3. Dan Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium, February 2004.
4. Paul Buitelaar and Philipp Cimiano, editors. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, 2008.

5. Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker. Ontology Learning. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 245–267. Springer, 2009.
6. William W. Cohen and Haym Hirsh. Learning the classic description logic: Theoretical and experimental results. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 121–133. Morgan Kaufmann, 1994.
7. Alexandre Delteil, Catherine Faron-Zucker, and Rose Dieng. Learning ontologies from RDF annotations. In Alexander Maedche, Steffen Staab, Claire Nedellec, and Eduard H. Hovy, editors, *Proceedings of the 2nd Workshop on Ontology Learning (OL) at IJCAI*, volume 38 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2001.
8. N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Concept formation in expressive description logics. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Proceedings of the 15th European Conference on Machine Learning (ECML)*, volume 3201 of *Lecture Notes in Artificial Intelligence*. Springer, 2004.
9. Gunnar Aastrand Grimnes, Peter Edwards, and Alun D. Preece. Learning meta-descriptions of the FOAF network. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, volume 3298 of *Lecture Notes in Computer Science*, pages 152–165. Springer, November 2004.
10. Sebastian Hellmann, Jens Lehmann, and Sören Auer. Learning of OWL class descriptions on very large knowledge bases. *International Journal on Semantic Web and Information Systems*, 5(2):25–48, 2009.
11. Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 Web Ontology Language Primer. Recommendation, W3C, October 2009.
12. Jeff Howe. The rise of crowdsourcing. *Wired*, 14(6), 2006.
13. Alexander Maedche and Valentin Zacharias. Clustering ontology-based metadata in the semantic web. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, volume 2431 of *LNCS*, pages 348–360. Springer, 2002.
14. Marta Sabou, Mathieu d'Aquin, and Enrico Motta. SCARLET: SemantiC RelAtion DiscoveRy by Harvesting OnLinE OnTologies. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *Proceedings of the 5th European Semantic Web Conference (ESWC)*, volume 5021 of *LNCS*, pages 854–858. Springer, 2008.
15. Katharina Siorpaes and Martin Hepp. Ontogame: Towards overcoming the incentive bottleneck in ontology building. In *Proceedings of the 3rd International Federation for Information Processing Workshop On Semantic Web & Web Semantics (SWWS '07)*, volume 4806 of *Lecture Notes in Computer Science*, pages 1222–1232. Springer, 2007.
16. Katharina Siorpaes and Martin Hepp. Games with a purpose for the semantic web. *IEEE Intelligent Systems*, 23(3):50–60, 2008.
17. Gerd Stumme, Andreas Hotho, and Bettina Berendt. Semantic web mining: State of the art and future directions. *Journal of Web Semantics*, 4(2):124–143, 2006.
18. Johanna Völker, Pascal Hitzler, and Philipp Cimiano. Acquisition of OWL DL axioms from lexical resources. In *Proceedings of the 4th European Semantic Web Conference (ESWC'07)*, volume 4519 of *Lecture Notes in Computer Science*, pages 670–685. Springer, Juni 2007.
19. Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the Special Interest Group on ComputerHuman Interaction conference on Human factors in computing systems*, pages 319–326. Association for computer machinery, 2004.