Computing and Informatics, Vol. 34, 2015, 652–677

SELF-ENRICHING ONTOLOGY-BASED CASUAL LEARNING GAMES

Nils Malzahn, Sabrina Ziebarth
Dominik Kloke, H. Ulrich Hoppe

Department of Computational and Cognitive Science
University Duisburg-Essen
Lotharstr. 63/65
47058 Duisburg, Germany
e-mail: {malzahn, ziebarth, kloke, hoppe}@collide.info

Abstract. Computer games are currently one of the computer science applications with the highest amount of users. "Serious gaming" approaches try to use the attraction of playing games to convey serious content in an entertaining way. This paper presents a multi-agent-architecture for collaborative, serious and casual games. We are combining serious games with casual games, as these are known to have a high potential for frequent gaming by people of various social and educational background. To be flexible concerning the learning domain an ontology-based approach has been used. Thus, new games for different knowledge domains can be created by exchanging the ontology accordingly. Furthermore, we use a "game with a purpose" approach to enable the computer to learn new facts about the games knowledge domain to ease the effort of adapting the basic game to the intended target domain. The presented system is capable of proposing candidates for missing relations in the underlying ontology as well as pointing out possible misconceptions – either within the ontology or by the players – using various heuristics including "wisdom of the crowds" methodologies. The feasibility of the presented approach and its implementation (the Matchballs game) is shown by three case studies each focusing on one aspect of the system.

Keywords: CSCL, multi-agent-architecture, serious games, games with a purpose, ontologies

1 INTRODUCTION

The "serious gaming" approach tries to use the attraction (i.e. the fun factor) of computer games not only for entertainment purposes, but also to convey serious content at the same time. Serious games have been established in vocational and advanced training over the last years and have a big potential for informal further vocational training [1]. While serious games are used for educating humans, in "games with a purpose" (GWAPs) (cf. [2]) human intelligence is exploited to facilitate tasks that are otherwise difficult or impossible to reach by computational means, i.e. the computer learns from humans ("human computing") (see also Figure 1).

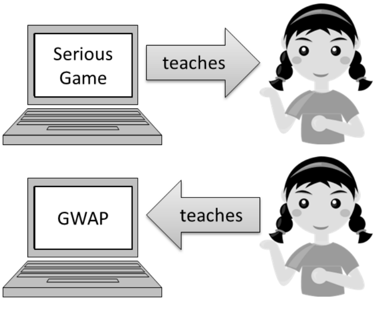


Figure 1. Serious games vs. GWAPs

Thus, apart from teachers and students being interested in getting information about common misconceptions to correct these, it should be possible to use the so called "wisdom of the crowd" to enrich a pre-built ontology. Consequently, regarding the GWAP context the game can be characterized as an ontology maintenance game.

Of course, such a game will not be a perpetual motion machine, since there has to be a knowledge base in the beginning ("stimulus energy"), and the gained "knowledge" should be revised by experts before providing it to learners ("energy loss by friction").

Our research investigates whether it is possible to create a motivating game that educates and gains new knowledge at the same time. In a way we try to transfer the "learning by teaching" [3] methodology to computers. We therefore combine the approaches of serious games [4] and "games with a purpose" [2].

1.1 Related Work

Serious games are often used to virtually train situational behavior like conflict resolution or firemen training or to implicitly transport some knowledge that would not be transferred easily otherwise, because it is too abstract (e.g. nutritional education for young diabetes patients) [4]. They have been applied to a broad spectrum of application areas, e.g. corporate and military training, healthcare, education and cultural training [5]. Some authors [6] define serious games as being inherently complex, but there are various other definitions of serious games [7] stressing the entertainment aspect [4].

GWAPs are typically casual games, in which pairs of players collaborate to reach the goal specified by the game. By playing the game, they externalize knowledge, which can be incorporated (learned) by the technical system. Typical fields of application for this "human computing" are audio, image, video and text annotation as well as ontology building and alignment[1] [2, 8, 9]. An example for a GWAP used for image annotation is *ESP Game*, in which two players have to guess what words the partner is using to describe an image [2]. The word they "agree" on is considered to be a good label for the given image and can be used to index it. *Verbosity* [8] is an example for a GWAP aiming at ontology creation and maintenance. Similarly to *Taboo*^{TM} one player takes the role of a "narrator" describing a secret word by filling sentence templates and the other player has to respectively guess the secret word. By describing the word the narrator states common-sense facts regarding this word. Another example for ontology creation GWAPs is *OntoPronto*, in which two players try to map randomly chosen Wikipedia articles to the Proton ontology [9].

While GWAPs could partially be used for training, e.g. ESP Game or Verbosity could be used by non–native English speakers to train their English vocabulary, this aspect is not discussed in the literature.

Casual games are known to be small games with a high potential for frequent gaming by people of various social and educational backgrounds. These games are characterized by simple and easy to learn rules and either by slowly increasing difficulty or a time limit combined with a high score list. They are not very time consuming and can be played occasionally, so they not only appeal to regular gamers but to the mass audiences [10]. Examples for casual games go back to games like Tetris and Pong and are currently produced for all types of devices including but not limited to mobile phones and especially browser games.

1.2 Game Idea

To create a game which educates users and learns from them at the same time, we decided for a simple allocation game, in which users link concepts to create simple statements (see Figure 2). A statement consists of two concepts linked by a predetermined relation types.

The game itself is kept simple with easy to learn rules and controls, and a session should only take a few minutes, like a "casual game", so it can be used in formal as well as in informal learning contexts.

Furthermore, the game is collaborative and playable either as a two player game or as a single player game with a bot. The goal is to agree with the teammate on as many relations as possible in a given time.

---

[1] see for example: `http://semanticgames.org`, last access: 20.08.2013

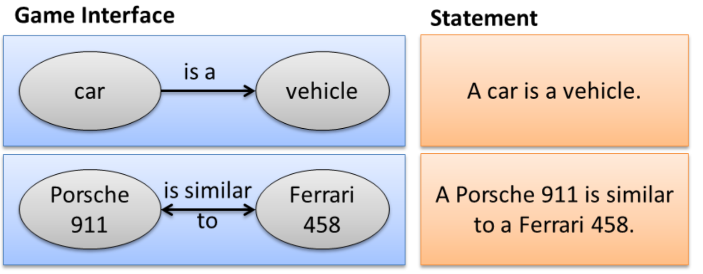


Figure 2. Game idea

1.3 Expected Benefits

In the multiplayer mode two players play together, share their knowledge and are rewarded for agreements. In the single player mode one player plays together with a bot which has the (correct) knowledge of the ontology, so this mode can be used for self-testing. Based on a constructivist view, the collaboration between the learners leads to knowledge sharing [11]. Furthermore, the opportunity to work together is supposed to be an incentive [12].

If the provided knowledge base is relevant to the learning context, students can use the game for repeating learning content (similar to a vocabulary trainer). The possible statements may either be concrete (like in case study 1 and 2) or they are more general (like in case study 3). If the knowledge base contains the main (higher) concepts of a topic, learners may be requested to abstract and generalize their knowledge.

The content of the games is provided as an ontology, so it can cover a wide range of domains. The ontology has very simple structure and thus can not only be created by knowledge engineers, but also directly by teachers (as excel sheet). Since ontologies are usually incomplete, because it is nearly impossible to represent even a limited domain in full detail [13], relations created by the players that do not occur in the knowledge base are not necessarily wrong, but possibly just missing, especially if a significant amount of players creates them. Therefore frequently occurring relations are very interesting, because either they can be used by know ledge engineers to enhance the ontology or they are typically misconceptions to be resolved by the teachers.

The overall process, the involved parties and resulting artifacts are displayed in Figure 3.

2 MATCHBALLS – GAME PLAY AND INCENTIVES

Based on the reasoning above the Matchballs framework for simple ontology based allocation games was created. In Matchballs the player creates statements by linking ("matching") concepts represented by balls (see Figure 4). A statement consists

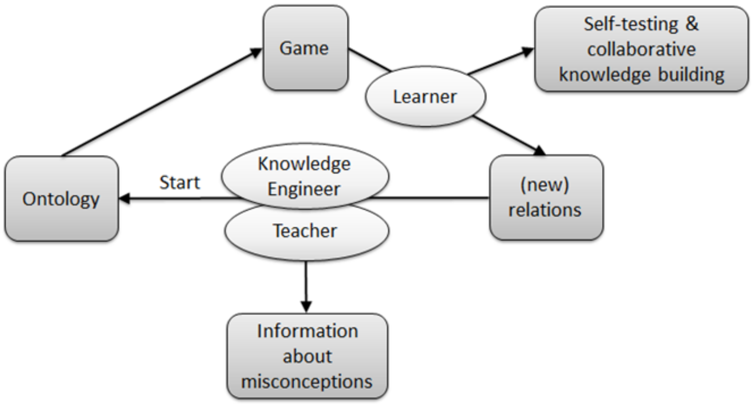


Figure 3. Benefits of the game for learners, teachers and knowledge engineers

of two concepts linked by one of four predetermined relation types. Each pair of players sees the same game field which contains 15 concepts. If the players agree on a relation, they score points and get time bonuses. They may see the relations of their teammates, but not the relation types. The relations still to be matched by the other player are visualized by hour glasses. The relations proposed by the teammate are shown as lines of question marks. If the teammates agree on a relation, this relation is displayed by stars, if they disagree, it is marked by red Xs.

In case study 1 we use the knowledge domain of food safety and hazardous material regulations, which is an important topic of further education in the German food industry. The considered concepts are specific situations, actions, dangerous substances and edibles, which can be linked by using the four semantic relations "is similar to", "is more general than", "results in" and "then you may not".

Feasible statements are for example:

- <Chicken> <is similar to> <turkey hen>
- <Machine overheats> <results in> <fire danger>
- <Oil starts burning> <then you may not> <extinguish the fire with water>

To motivate the players to play the game, several incentives considering different types of players are included into the game. For competitive players there are high scores and time bonuses, which are well-known and often used incentives (e.g. cf. [14]) since early arcade games. Furthermore, players can collect "achievements", which are trophies for solving certain predefined tasks. In Matchballs achievements are for example titles for playing a given number of games with another player ("team player") or with the bot ("robot's friend"), or cups for gaining certain amounts of points or extending the game for certain time spans (see Figure 5). Achievements are a more recent kind of incentives often used in modern console games. They not only address competitive players, but also people with collector's passion, who

Figure 4. Matchballs user interface

want to unlock the full set of obtainable awards. While competitive players tend to play against the bot to be not dependent on the teammate, for team players the possibility to play together with another human is an incentive of its own (cf. [12]).

3 ARCHITECTURE

The Matchballs system is based on a multi-agent architecture. "The concept of multi-agent architectures relies on the idea that a collection of autonomous processes, called agents, can achieve intelligent problem-solving behavior by coordinating their knowledge, goals, skills, and plans" [15]. This kind of architecture is especially useful for systems which integrate different processing and reasoning methods and have the possibility to divide the problem solving knowledge into independent pieces [15]. Furthermore, multi-agent system can be easily extended.

These characteristics fit our goals very well. While there are currently only few analysis agents, there are lots of methods, e.g. from statistics, data mining or information retrieval, which could be used for further analysis integrating different theories. Furthermore, the results of our first tests (see Section 4.1) could be used for the development of further agents.

The central game server is based on the *SQLSpaces* [16] tuplespace implementation. Tuplespaces are inspired by blackboard architectures [17], which are characterized by a data-driven approach. Their general principle is to have no direct

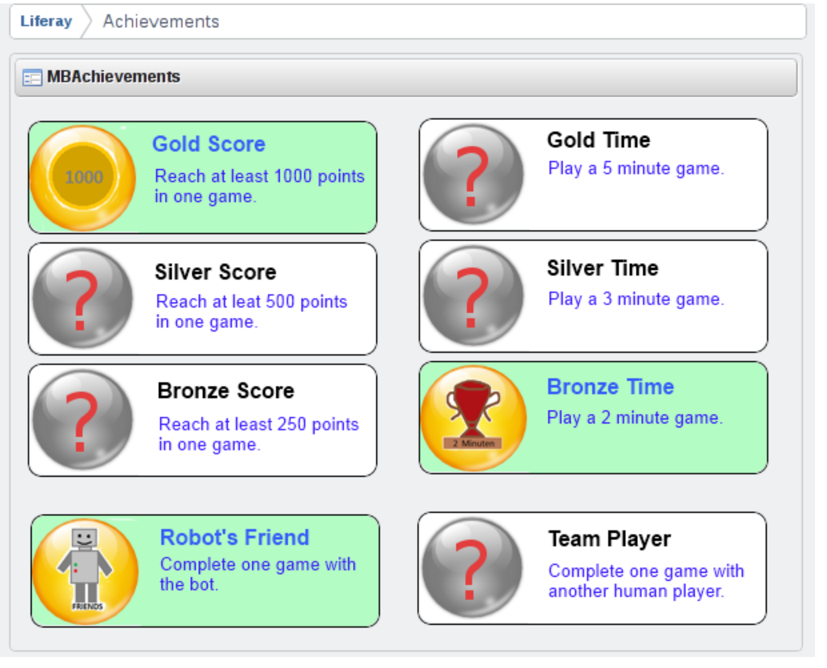


Figure 5. Display of achievements

communication between the agents. These agents communicate by writing and reading messages on and from the blackboard (i.e. the tuplespace). These messages consist of tuples made of primitive data types (integer, characters, booleans) and strings. This allows a programming language heterogeneous approach, since these data types are available in almost all programming paradigms and languages. A single tuplespace server may contain several tuplespaces used to divide the data stored in the server into logic or semantic units.

The Matchballs architecture distinguishes five different categories of tuplespaces: *Configuration Space*, *Coordination Space*, *Game Spaces*, *Intermediate Space*, and *Ontology Space* (see Figure 6).

The *Configuration Space* holds the necessary information about the basic setup of the game, i.e. which subset of four relations from the ontology residing in the *Ontology Space* is available to the players. Additionally it is possible to enter labels that are presented to the players for these relations and eventually change the direction of a specific relation. This was included, because the nomenclature of the ontology may be too confusing for an average player. For example we had to change the initial label of a relation “more general than” to “more specific than”, because for some reason this was easier for the subjects to decide. Of course, in this case we did not change the ontology, but just the labels and the subsequent interpretation of the played relations.

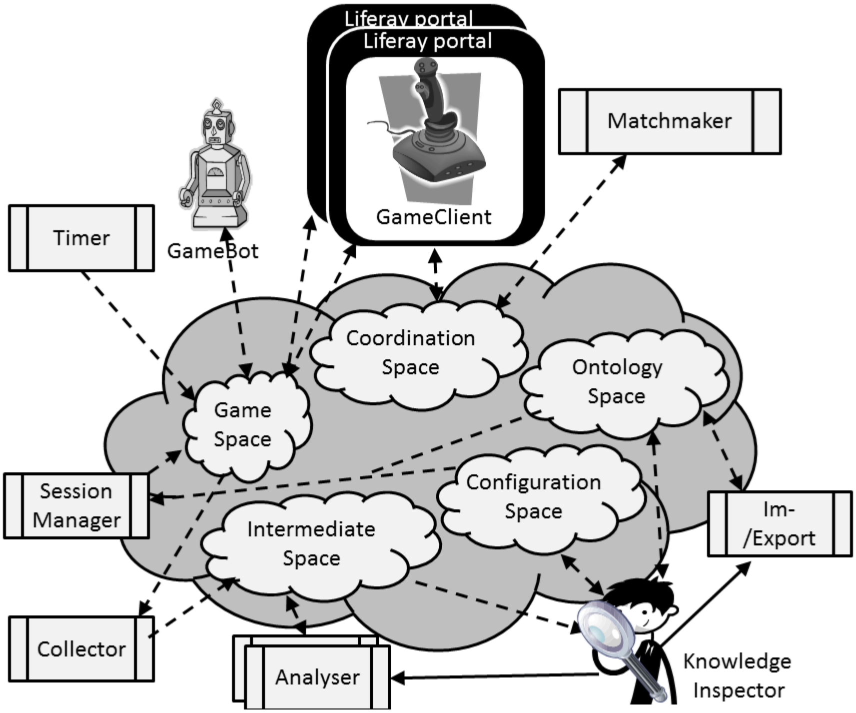


Figure 6. Architecture of Matchballs

The *Coordination Space* is used to conduct the matchmaking between two human players or to start a single player game. The GameClients register at the *Coordination Space* to announce their availability for a new game session and retrieve the information about the *Game Space* they have to connect to. In our case studies the GameClients reside in a Liferay[2] portal, which is the JSR 2.0 portlet-oriented enterprise portal that is used for the web presence of FoodWeb2.0; the project where the case studies are embedded (see Section 4). The GameClient is implemented only using HTML5 and JavaScript to cover most of the current browsers on the major operating systems including those that have no flash support. Furthermore, this allows any HTML5-supported content inside of a Matchballs. Although currently only images and text are used in a ball, video and audio files could also be played, e.g. for process sequencing tasks.

The first implementation of the agents was completely in Java [18]. The current implementation additionally makes use of a prolog interpreter, which is used to calculate analysis rules formulated in Prolog's horn logics. A further explanation of

[2] `http://www.liferay.com/en/`

this mechanism can be found in Section 3.4. In the next sections we will explain the core components of the architecture in more detail.

3.1 The Game Space

Each game session has its own *Game Space*, to which either two human players – put together by the Matchmaker agent through the *Coordination Space* – for multiplayer games or a human player and a GameBot for single player games are connected. The *Game Space* holds all necessary information for a Matchballs game. That means the *Game Space* consists of an excerpt of the *Ontology Space*, the current timer as set by the Timer agent, the current score, and the links that have been made by the players as well as its assessment by the Session Manager. The GameBot has access to the whole information stored in the ontology excerpt, i.e. it is aware of the complete knowledge that is represented in the respective ontology excerpt. Thus, all associations made by the GameBot are correct assuming that the ontology is adequately modeled. The excerpt from the ontology is created by the Session Manager agent. It takes care that there is always a minimum of possible relations between balls in the beginning. It also detects concordances of the two players with respect to the links between balls made by each player. If an agreement on an association is detected by the Session Manager, i.e. both players' GameClients wrote exactly the same tuple representing a link between two balls to the *Game Space*, the Session Manager updates the score tuple of the current *Game Space* and writes an assessment tuple for this link into the Game Space. A game ends if the time is up. Afterwards the links made by the players (regardless if it was a multiplayer or single player game) are collected by the Collector and put into the *Intermediate Space* for further inspection. The *Game Spaces* are discarded in a productive environment to save space. At the moment the Collector just counts the occurrence of the specific relations made by the players and stores or updates the amount in the *Intermediate Space*, because further analysis is conducted on the data within the *Intermediate Space*.

3.2 The Ontology Space

The *Ontology Space* holds a tuple representation of the ontology. Every concept and relation is represented by a tuple. The markers set by the analysis agents are also represented in the ontology. The ontology design is based on SKOS (Simple Knowledge Organization System) [19]. SKOS is quite a simple representation mechanism, but it is sufficient for our game. In our case study, we distinguish only four association types, easily derived from SKOS-relations and we have few concepts/classes and many individuals/instances, which can be categorized in SKOS's concept schemes. The restrictions on the ontology are caused by the spirit of the game. Since it shall be a casual game for the masses, a huge, sophisticated system of associations and concepts/classes would be misleading. With respect to the use of the game as an ontology enrichment game, it is also not feasible to have a complex representation scheme. Last but not least, there is a plugin for Protégé for SKOS. Thus, a popular editor for ontology creation, inspection and refinement can be used. The import-export agent takes care of the proper translations of the tuple format to SKOS (OWL format) and vice versa. The ontology used in case study 1 (see Section 4.1) consists of 191 individuals connected with 91 associations distributed on the different association types. The structure of the ontology is displayed in Figure 7.

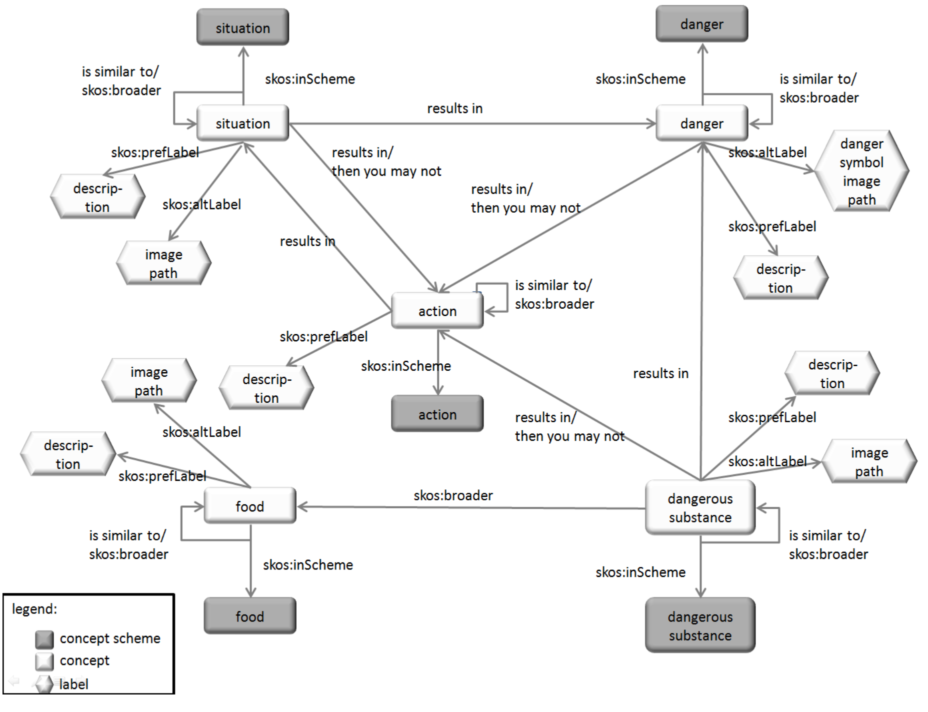


Figure 7. Ontology structure – examples taken from case study 1 (Section 4.1)

3.3 The Intermediate Space

The Intermediate Space is used for analyses. The first implementations of Matchballs [20, 18] conducted only basic analyses. Either a threshold of at least five different players linking two concepts with the same association type (not present in the ontology yet) to propose this association for inclusion into the ontology or frequent contradictions to the modeled ontology by the players' links were used to detect possible misconceptions.

The current system employs an increasing set of heuristics to assess the relations found in the *Intermediate Space*.

3.4 Analysis Agents

All heuristics are encapsulated in agents independently working on the tuplespace. Currently the agents are written in Java or (SWI) Prolog. Actually the Prolog-Agent employs a meta-interpreter to evaluate rules specified by a *Knowledge Inspector* (see Section 3.5) user. At the moment rules have to be specified as valid Prolog rules. In the future these rules may also be specified by simple "if-then-else" rules to ease the creation of these rules.

To restrict the amount of relations to be presented to the knowledge interface user, only those relations are considered that have been expressed by minimum number of different players. The specific amount can be set in the *Configuration Space*, e.g. within the *Knowledge Inspector*.

The further inspection of the remaining data is conducted based on rules, which evaluate first the significance of a relation and second determine the type of the observed phenomenon. Currently, two types of phenomena are distinguished:

1. Ontology gaps, i.e., a particular relation is currently not part of the ontology and should be added.
2. Misconceptions, i.e., a particular relation is either not part of the ontology and must not be added. Variants of this type indicate that there is a relation between those two concepts, but the selected name is wrong or just the direction of the relation is wrong.

Five different heuristics (each implemented as an autonomous agent) are considered if a relation is created. Depending on the respective heuristic a relation may be classified as a gap, a misconception or both.

1. Only relations that have been played by at least a minimum number of different players are considered at all (threshold condition). The relation must not already be part of the ontology. This ensures that a new relation has a minimum support. Therefore it should be normalized with respect to the overall number of players.
2. If the relation has met condition 1, it is checked if the given relation with reversed direction exists between the given concepts from the played relation. If this is the case, the relation is shown to the knowledge interface user as a possible misconception or error in the ontology:

   (a) if the reversed relation has already be confirmed (as indicated by a special tuple) the relation is only classified as a misconception.
   (b) if the played relation has already been classified as wrong, the played relation is only classified as a misconception.
   (c) if the reversed tuple has not been classified yet, the *Knowledge Inspector* user is asked to either confirm the new relation and add it to the ontology, to replace the existing reversed relation by the new relation (classifying the old one as "wrong") or just to classify the new relation as wrong and therefore

classify the new relation as a misconception. Until the user decides the new relation is classified as both "gap" and "misconception".

3. If the relation has met condition 1, it is checked if the given two concepts have a directed link with each other with the same direction at all. If there is another relation, the new relation is analogously classified to 2a–2c.
4. An ontology may define transitive relations. If a played relation meets condition-1, but there is a path along the same transitive relation between those two concepts, the relation is considered correct and missing from the ontology. Thus, it is added to the ontology without further notice of the knowledge interface user. Thus, the correctness does not have to be validated again in subsequent analysis runs and the transitive hull does not have to be calculated and inserted in advance. Symmetric relations are handled analogously.
5. An ontology often defines a relation at the highest possible position, e.g. if there is a kind of generalizing relation between concepts A and B ($A \Leftarrow B$) as well as between C and D ($C \Leftarrow D$) and another relation between A and C ($A \leftarrow C$), the following relations should also be valid, although they are not explicitly part of the ontology's set of relations: $A \leftarrow D$, $B \leftarrow D$. So the relation $\Leftarrow$ allows "replacing" one concept for another. Not only formal "generalizing" relations like "is-a" or "subclass-of" may be marked as "replacing", but also other relations like e.g. "similar-to". The last example indicates that relations that are most likely correct, may also be wrong, so these new relations are presented as possible additions to the knowledge interface user, but not automatically added to the ontology. It is also noted that this type of replacement is not embedded in the regular reasoners (like the transitive hull).

Depending on the use case of the game, relations that have already been marked "wrong" may be filtered to avoid showing these relations to a knowledge engineer who is solely interested in ontology improvement, more than once. For a teacher the misconception is still relevant, because in this case the students are the target for improvement.

3.5 Knowledge Inspector

The *Knowledge Inspector* (see Figure 8) provides a web based user interface for knowledge engineers and teachers to Matchballs game settings and the analysis results of the agents.

It is completely written in HTML5 and Javascript using a web socket connection to access the SQLSpaces. User actions in the *Knowledge Inspector* result in specific command messages in the tuplespace that are processed by agents (like an OWL-Importer) listening for this kind of tuples.

As shown in Figure 8 the *Knowledge Inspector* can be used to set the threshold for considering a relation (see condition 3.4 (1)) as well as basic ontology editing by creating new concepts and relations. Additionally existing relations may be

Figure 8. Knowledge Inspector – overview and selection of game relations

edited and specific relations may be defined as "replaceable" in the sense of condition 3.4 (5).

Although it is advisable to use a sophisticated ontology editor like Protégé, the functionality to make a small change without exporting and importing the whole ontology has been proven worthwhile.

The game ontology can be imported and exported to SKOS's OWL/RDF-representation using a Java agent based on JENA libraries. Additionally a basic ontology can be imported based on a textfile where each line defines a relation between two concepts. This representation allows teachers, who usually have no detailed knowledge about ontology engineering and formalisms, to build a meaningful ontology for their students.

The initial reason to build the *Knowledge Inspector* is accessible by clicking on "analysis results". The results are prepared in three categories:

- Possible new relations
- Possible misconceptions
- Results ordered by analysis method

The first two filtered views are intended for all users of the *Knowledge Inspector*. The second one is especially useful for teachers who want to know about the errors made by their students. Selecting either one of these views leads to an interface where every addition/misconception is listed and the expert may choose between confirming or rejecting it (see Figure 9). Since heuristic 3.4 (1) is included in heuristics 3.4 (2a) and (3) the results of (1) are only shown, if (2) and (3) do not classify the relation.

A confirmation of an “new relation” proposal leads to the integration of the respective relation to the ontology. A reject marks this relation as “wrong” for further processing with (other) heuristics. Rejecting a “misconception” means that the relation is in fact a missing relation in the ontology. Thus, the user is asked if it should be added. A confirmation just acknowledges it for training purposes of the heuristics.

Complementing the other view the third view addresses developers of heuristics. It relates the results of the respective heuristic with the decision made by the experts in the other two views. Thus, the heuristics can be evaluated with respect to their predictive capability.

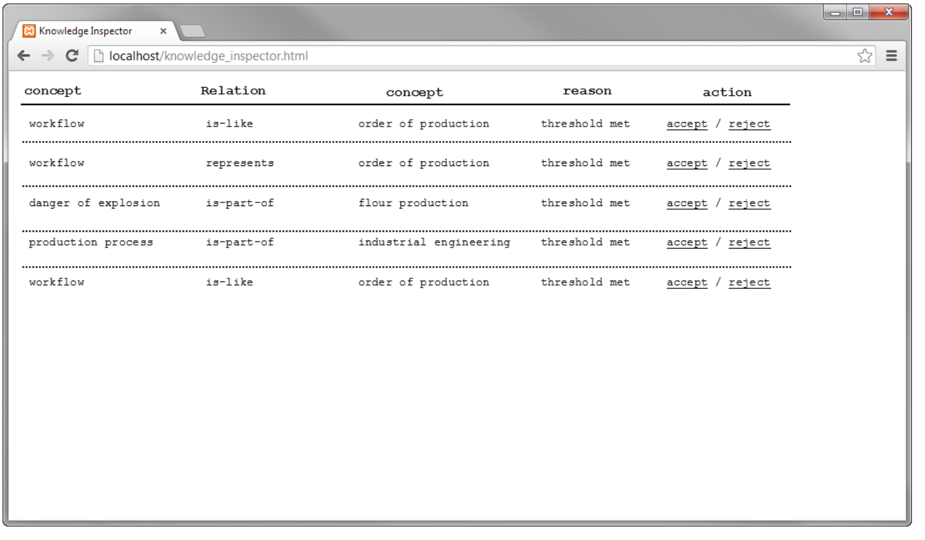


Figure 9. Knowledge Inspector – evaluating analysis results

4 CASE STUDIES

To evaluate the soundness and usefulness of our approach we conducted three case studies. The first study focused on the soundness and usefulness of the game Matchballs as a means to improve a given ontology by learners in the domain of the ontology model. Thus, the heuristics were broadly replaced by human expertise and manual categorization. The second case looks into the usefulness and appropriateness of the Knowledge Inspector. While the first two case studies where applied to (more or less) controlled environments, in the third scenario the game was freely played in the context of an exam preparation.

The first two case studies were conducted with members of the German food industry embedded in a nationally funded project. The project Foodweb2.0 (funded by the German Ministry of Research and Education) aims at training the employees of the German food industry using two basic strategies: motivating employees for vocational training and performing education in collaborative, blended learning using Web2.0 technologies.

The German food industry is characterized by a high amount of workers without or with only a low level of formal qualification. While these workers are easily found and taught to perform the simple and often physically exhaustive tasks, there is a lack of employees with a higher qualification (e.g. skilled workers), who are able to use and control the complex machines and processes of the food production industry. Thus, the human resource managers try to train some of the low qualified to a higher qualification level to close the gap. This is not an easy task since these people often have a migratory background and therefore language problems and/or they are not very motivated to learn because of various reasons, e.g. education is not an asset to them or their work is so exhausting that they are not ready to learn.

Our framework is well suited to support this scenario as it combines both strategies. While playing the game the players have to remember facts and rules in the context of the respective domain modeled with the ontology. Since it is an online game it can be used for training and recapitulation, not only in a course, but also at home. The motivational properties of a casual game should help to tunnel the barriers between learning and exhaustive daily routines.

4.1 Case 1 – Evaluating the Game

For this case study we created an ontology on the domain of food safety and hazardous material regulations. The structure of this ontology is shown in Figure 7. The Matchballs game has been evaluated with a class of 18 students at the Academy of Sweets in Solingen, Germany. Since there were only six laptops available for conducting the experiment, after a short plenary introduction of the game, the class was divided into three groups of six students. The students of each group had a time-slot of five minutes for playing as many games as possible and were encouraged to start with a single player game for learning the controls and then perform at least one game with a human partner. After playing the game they had to complete a questionnaire and at the end there was a short plenary discussion.

The subjects consisted of five females and thirteen males, sixteen were German native speakers. They rated themselves to have high knowledge in the fields of safety at work and danger symbols (median of five on a scale from one to six) and also some knowledge of hazardous materials (median of four on a scale from one to six). 66.67 % declared to have participated in a course on safety at work and 55.56 % on a course on hazardous materials. Thus, the participants are considered to have at least basic knowledge of food safety and hazardous materials and to represent a group recapitulating their knowledge on these topics.

Together the subjects created 155 different relations, most were only created by one user, but there were also relations created by up to seven different participants (see Figure 10). The most often created relations are displayed in Table 1.

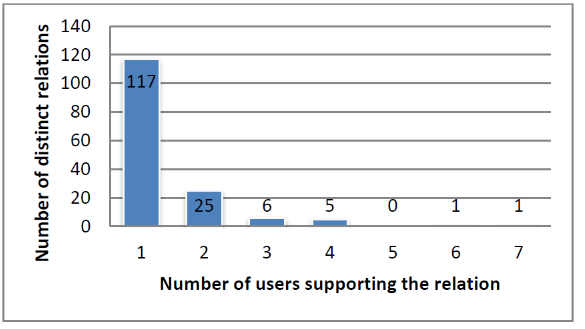


Figure 10. Histogram of relations

The higher the amount of users supporting a relation, the higher is the probability that the relation is correct (precision) (see Table 2). A frequent error lies in choosing the wrong direction of the relation, around 26 % of the overall relations are correct except for the direction; if not all relations but the ones with a minimal support higher than one are considered, the percentage is even higher (see Table 2). These errors can be considered as careless mistakes instead of real misconceptions. Because of these mistakes, it is not amazing that the undirected relation "is similar to" has a better overall precision (0.53) than the other directed relation types. The relation "is more general than" is chosen most seldom (12.9 %) and has the worst overall precision (0.3). In the questionnaire 83.3 % of the subjects stated that they did not understand the meaning of this relation. The discussion in the plenum showed that flipping the relation and labeling it "is more specific than" would lead to a much better understanding, thus the relation was exchanged in the current version of the game. With respect to the other relation types the subjects stated that they could comprehend them (94.4 % the relations "is similar to" and "then you should not" and 88.9 % "results in").

The frequent correct relations having a support of at least three (see Table 1) already are in the ontology. This is due to the initial composition of the playing field containing two thirds of concepts being linked in the ontology and only one-third of concepts being not linked with the other selected concepts (see Section "Architecture"). But these relations only cover 17.7 % of the correct relations created by the users. Thus, the relations with high support have a high precision, but a low recall. 75.8 % of the correct relations already are in the ontology, 4.8 % of the correct new ones are supported by two and 19.4 % by only one user. The low support of the correct new concepts on the one hand can also be explained by the initial composition of the playing field and on the other hand by the limited number of games played in the experiment.

The users show different profiles of creating relations considering the number of different relations created, their precision, the number of correct relations missing in the ontology and the number of errors based on direction issues (see Table 3).

concept	**Relation Type**	**Concept**	**Support**	**Assessment**
pork	is similar to	beef	7	correct
pear	is similar to	apple	6	correct
switch on machine	is similar to	switch on computer	4	correct
barley	is similar to	rye	4	correct
machine uses laser beams	results in	danger by optical radiation	4	correct
machine shows functional disorder	results in	switch off machine	4	correct
radioactive material escapes	results in	radioactivity	4	correct
close the window	results in	draft at place of work	3	incorrect
danger of explosion	results in	explosive materials are processed	3	incorrect
spelt	is similar to	barley	3	correct
chicken	is similar to	turkey hen	3	correct
machine is on high-voltage	then you may not	splash with water	3	correct
machine overheats	results in	fire danger	3	correct

Table 1. Relations support by the biggest amount of different users (translated to English)

Minimal Support	**Precision**	**Errors Based on Wrong Direction**
4	1.00	none
3	0.85	100.00 %
2	0.68	58.33 %
1	0.40	25.80 %

Table 2. Comparison of minimal support for considered relations, precision of the relations and percentage of errors based on wrong direction of the relation

Each user on average created twelve relations, of which six (0.51 %) were correct, one (8.3 %) was new and correct and ca. two (15 %) were wrong because of direction problems. There is no correlation between these variables. Partitioning the students into two clusters using the well-known kMeans algorithm [21] (see Table 4), reveals one small group of students (22.2 %), who create a number of distinct relations far below average, are a bit more precise and make no direction mistakes and a larger group, who create a bit more distinct relations than average, are a bit less precise than the other group and making much more direction errors.

User	Distinct Relations (Activity)	Precision (Quality)	New Correct Relation (Innovativeness)	Direction Mistakes (Sloppiness)
A1	12	0.50	8 %	17 %
A2	4	0.25	0 %	25 %
A3	17	0.47	12 %	12 %
A4	23	0.43	4 %	22 %
A5	14	0.57	7 %	29 %
A6	8	1.00	13 %	0 %
B1	6	0.33	17 %	0 %
B2	7	0.29	0 %	0 %
B3	10	0.60	0 %	30 %
B4	12	0.50	0 %	17 %
B5	8	0.50	13 %	25 %
B6	11	0.55	0 %	18 %
C1	8	0.63	13 %	13 %
C2	16	0.56	13 %	31 %
C3	18	0.67	6 %	6 %
C4	0	–	0 %	0 %
C5	26	0.27	8 %	8 %
C6	18	0.61	22 %	6 %

Table 3. User profiles

	Cluster 1 (14 students)	Cluster 2 (4 students
Distinct Relations	14.07	5.25
Precision	0.49	0.51
Innovativeness	0.07	0.07
Sloppiness	0.14	0.00

Table 4. Cluster centers

In general the students had fun playing the game and experienced it as motivating (each variable had a median of four on a scale from one to six). 77.8 % stated they would like to play the game again and on average they would play it once to several times a month. The complexity is perceived as medium (median of 3.5 on a scale from one to six), which is an indicator that the game is neither overstraining nor boring and hence appropriate.

4.2 Case 2 – Knowledge Inspector

The ontology of the second case study models knowledge from the domain of food engineering knowledge that is relevant for the final exam of food engineers.

Eleven subjects (23–34 years old) that had no experience within the food engineering field played the game to generate game data. The educational level of the subjects was heterogeneous; from no educational attainment to academic degree.

The ontology consists of 178 concepts with 128 relations linked by four relation types: *is-like*,*is-part-of*, *is-not*, and *represents* (see also Figure 8).

The subjects were asked to play the game in phases. In the first five minutes the subjects were allowed to have a look at an extensive list of possible relations within the ontology. Afterwards the subjects played the game without any hints. The first phase allowed the subjects to get a first impression of what might be a valid relation in the domain of food engineering.

The subjects created 922 different relations by playing the game. The threshold for considering a relation was set to 5. With this setting two of the heuristics proposed 15 relations to be investigated: the "new relation" agent detected 3 hits and the agent detecting swapped relations had 12 hits. To get results from additional heuristics the threshold had to be lowered to 2 due to the low number of players, which in turn leads to a higher false-positive rate for the other two agents (see Table 5). Heuristics 4 and 5 did not produce any results, since the ontology does not contain a transitive or symmetric relation and no "replaceable" relation was marked by the experts.

Threshold	**Agent/Heuristic**				
	1	2	3	4	5
1	10	4	43	0	0
2	5	2	22	0	0
3	5	0	20	0	0
4	3	0	13	0	0
5	3	0	12	0	0

Table 5. Analysis of agents' results with different thresholds

The results of the agents were independently shown to two experienced teachers who are educating students becoming food engineers. The Matchballs game and its context was explained to the experts to clarify the source of the statements they had to evaluate in the next steps. Afterwards the experts were asked to identify misconceptions of "their" students based on the game data as well as finding and correcting errors in the ontology using the Knowledge Inspector. Finally an expert interview concerning the usability and the soundness of the heuristics was conducted.

Although the usability should be improved, because the experts were not completely satisfied with the navigation through the system, the tasks could be solved with ease. Looking at the soundness of the heuristics the current results are ambiguous. The two proposed additions to the ontology were accepted unanimous, but the opinions about the misconceptions were mixed. One expert accepted six of the twelve proposed misconceptions as ontology errors and wanted to correct these and the second one accepted eight of the misconceptions and corrected four relations of the initial ontology due to this interpretation.

4.3 Case 3 – Informal Context

In the third scenario Matchballs game was applied to a master level university course on *modelling of interactive learning systems* for exam preparation. The goal was to give the students an opportunity to reflect on the relations between the general concepts regarding the lecture topics. The game was integrated into the LMS (Moodle), thus students could play it freely on their preferred devices. A short questionnaire was linked to measure the perceived usefulness as well as the usability of the game. The used ontology was created by two exercise instructors of the course and contained 126 statements regarding the overall topics of the course.

The course consisted of around 40 students who took part in the lectures and did their homework regularly. 19 of them played the game and 8 also completed the questionaire. The amount of played games varies from 1 to 34, with an average of around 8 games per user and a standard deviation of 7.85 (see also Figure 11).

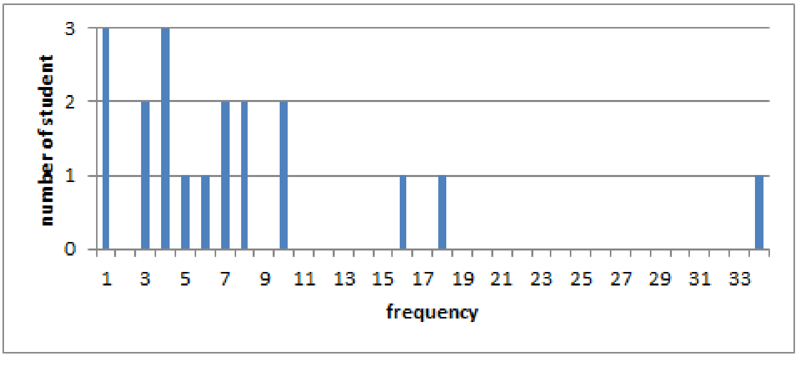


Figure 11. Histogram of the number of played games per user

Most games were played singleplayer, but there were also two multiplayer games involving three different players. Overall 454 distinct relations were played, 99 of them already were in the ontology and 355 not. Most of the new relations were only played one or two times (see Figure 12).

Regarding the analysis with the Knowledge Inspector we only considered relations occurring at least four times resulting in seven new relations, which were all true. Neglecting the threshold of four occurrences we could also identify eight relations fulfilling the transitivity condition (c.f. Section 3.4) (one overlapping with the threshold condition), which were all true. Thus, regarding the GWAP concept, we could identify new knowledge for our system.

The question, if the students learned something, is not so easily answered. Since only eight filled the questionnaire, the number of students assessments is very low (see Table 6). 75 % of the answers indicate that Matchballs is appropriate for recapitulation of learning contents and 50 % of the students partially agree that they learned new relationships regarding the learning content.

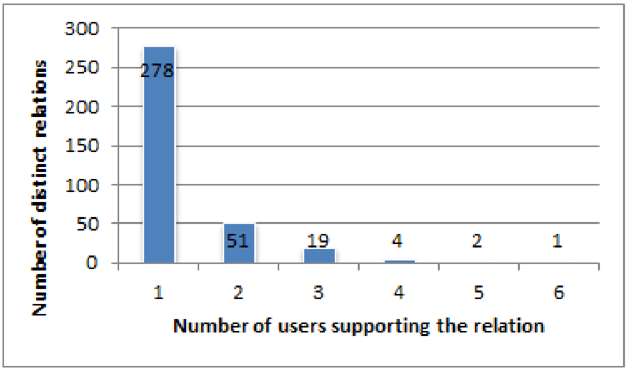


Figure 12. Histogram of relations

Question	Agree-ment	Partial Agree-ment	Neutral	Partial Dis-agree-ment	Dis-agree-ment
By playing Matchballs I could recapitulate contents of the lecture.	2	4	0	1	1
By playing Matchballs I learned new relations regarding contents of the lecture.	0	4	1	1	2

Table 6. Perceived usefulness

The exam results show that the students using the game were slightly better (average grade of 1.2) than the ones not using the game (average grade of 1.5). Within the group of students who played the game the ones who played it more often also tend to have better grades than the ones who only played it only one to three times. Of course, more motivated/hard-working students often tend to use additional opportunities for learning more, tend to get better grades, which might explain the above average grade results.

5 DISCUSSION

The evaluation results of the case study 1 show that the Matchballs game was perceived as a casual game that is "addictive" enough to encourage the learners to try another game to improve their score. Based on the data generated during the game sessions we were able to identify 17 relevant associations that were not represented in the initial ontology in the case study 1. These associations were integrated into the ontology by our knowledge engineers. Consequently, we could indicate that our approach of using a learning game also as "game with a purpose" is feasible. In this way the game may be seen as self-extending with respect to closing gaps in the ontology. This result is also confirmed by case the studies 2 and 3. The applied heuristics for detecting new relation and misconception candidates seems feasible.

The "wisdom of the crowds" approach is often criticized arguing that expert contributions would be enough. In the case study 1, the four "best" students (22.2 %) who created the biggest amount of correct new relations could only provide 46.67 % of the overall number of correct relations. This indicates that it is not only possible to retrieve valuable knowledge from less knowledgeable students, but also that including non-experts provides more additional knowledge than restricting the knowledge retrieval to experts.

At the moment the game can only be used for adding new associations in the ontology. In the future we plan to use the game to acquire new knowledge fields in a class session. The teacher may introduce a set of new concepts into an ontology which are not connected at all. Students are asked to play the game so that they will connect the new concepts with each other as well as with the old ones, therefore integrating them into the existing ontology. In a way the game may be viewed as a concept map creation game when used as a multiplayer game, where the players create a shared concept map. Concept maps are successfully used as a learning tool for linking the existing and new knowledge as well as for evaluation and identifying valid and invalid ideas of students [22].

Although, the effect of collaborative relation creation for learning is ambiguous [23, 24, 25, 26] and seems to depend on additional factors, the collaboration clearly does not impair the learning effect. Thus, a positive effect for the individual learner can be expected and the collaborative mode additionally may cause better performance and an enrichment of the underlying ontology.

If the game is played in a single player mode, the game may still be used as an advanced vocabulary trainer. In spite of playing the game individually the students still collaborate indirectly. Accordingly, teachers can apply the game to get an overview of typical misconceptions of the group but also of single students.

The introduction of a high score table may raise the question if it causes any friction among work or learning mates. As already mentioned above high score tables target people who are motivated by competition and may lead to unintended results (see [14]), nonetheless using well designed awards, like it is possible with Matchballs, can be used to keep competition on a reasonable level.

6 SUMMARY AND OUTLOOK

This article presents the Matchballs–System. It combines elements from casual games, serious games and games with a purpose. The presented approach has the potential to support learners and ontology engineers at the same time in various domains. The learning context provides a purpose that is beneficial for the learners and the results of the game play can be used to close gaps in pre-defined ontology as can be seen in the case studies 1 and 2. The students in the case studies 1 and 3 said that they would like to play the game again because it was perceived as entertaining and instructive at the same time. Additionally, we could show (cf. case study 2) that the use of the knowledge inspector is considered to be a helpful tool for teachers reviewing the learning progress of a class and for knowledge engineers revising an ontology.

The third case study shows that it is easy to exchange the learning domains (from "hazardous material" to "design criteria of computer supported learning systems") and the feasibility of the casual gaming approach for learning since the subjects liked playing the game.

In the context of the Foodweb2.0 project there have already been several requests by teachers and students for transferring the game to further knowledge domains. We will try to incorporate these domains and enhance these ontologies with specific feedback on the relations made by students. The feedback will be given at the end of the game. For the multiplayer scenario there will be a feedback about the existence of their relation in the ontology. In the single player scenarios the feedback hints at possible misconceptions automatically, based on information directly represented in the ontology for the particular error types and exploiting the semantic ontology structure for the generation of generic feedbacks like it is done in intelligent tutoring systems. A first step to domain independent tutoring can be the identification of players that contradict themselves by playing e.g. non-symmetric reversed relations. This behavior indicates uncertainty with respect to the correct semantics of either the particular relation or the involved concepts.

The second exploratory study showed that the Knowledge Inspector might be a useful tool for teachers and knowledge engineers to evaluate the results of a Matchballs game and to improve or adapt the existing ontology.

In summary, the presented framework can be used for both the ontology learning and the subject learning; and even for both tasks at the same time. That provides an added value for both the provider of a game (an ontology) and the students playing it.

REFERENCES

[1] Marr, A. C.: Serious Games für die Informations- und Wissensvermittlung. Number 28 in Bibliotheken auf neuen Wegen. Dinges & Frick GmbH, 2010.

[2] von Ahn, L.: Games with a Purpose. IEEE Computer, Vol. 39, 2006, No. 6, pp. 96–98.

[3] Gartner, A.—Kohler, M. C.—Riessman, F.: Children Teach Children: Learning by Teaching. Harper & Row, 1971.

[4] Michael, D.—Chen, S.: Serious Games: Games That Educate, Train, and Inform. Thomson Course Technology PTR, Boston, 2006.

[5] Rego, P.—Moreira, P. M.—Reis, L. P.: Serious Games for Rehabilitation: A Survey and a Classification Towards a Taxonomy. 2010 5$^{\text{th}}$ Iberian Conference on Information Systems and Technologies (CISTI), 2010, pp. 1–6.

[6] Westera, W.—Nadolski, R. J.—Hummel, H. G. K.—Wopereis, I. G. J. H.: Serious Games for Higher Education: A Framework for Reducing Design Complexity. Journal of Computer Assisted Learning, Vol. 24, 2008, No. 5, pp. 420–432.

[7] Susi, T.—Johannesson, M.—Backlund, P.: Serious Games: An Overview. Technical report, Institutionen för kommunikation och information, 2007.

[8] von Ahn, L.—Kedia, M.—Blum, M.: Verbosity: A Game for Collecting Common-Sense Facts. Proceedings of CHI 2006, 2006.

[9] Siorpaes, K.—Hepp, M.: Games with a Purpose for the Semantic Web. IEEE Intelligent Systems, Vol. 23, 2008, No. 3, pp. 50–60.

[10] Kuittinen, J.—Kultima, A.—Niemelä, J.—Paavilainen, J.: Casual Games Discussion. Proceedings of the 2007 Conference on Future Play, ACM, 2007, pp. 105–112.

[11] Cohen, E. G.: Restructuring the Classroom: Conditions for Productive Small Groups. Review of Educational Research, Vol. 64, 1994, No. 1, pp. 1–3.

[12] Johnson, R. T.—Johnson, D. W.: Using Cooperative Learning in Math. In: Davidson, N. (Ed.): Cooperative Learning in Mathematics: A Handbook for Teachers, Addison-Wesley, 1990.

[13] Hepp, M.: Possible Ontologies: How Reality Constraints the Development of Relevant Ontologies. IEEE Internet Computing, Vol. 11, 2007, pp. 90–96.

[14] Toups, Z. O.—Kerne, A.—Hamilton, W.: Motivating Play through Score. ACM Computer Human Interaction 2009 Workshop on Engagement by Design, 2009.

[15] Kipper, B.: Interlocking Multi-Agent and Blackboard Architectures. Proceedings of EPIA '95, 1995, pp. 371–375.

[16] Weinbrenner, S.—Giemza, A.—Hoppe, H. U.: Engineering Heterogeneous Distributed Learning Environments Using Tuple Spaces as an Architectural Platform. Proceedings of the 7$^{\text{th}}$ IEEE International Conference on Advanced Learning Technologies (ICALT 2007), Los Alamitos, CA, 2007, pp. 434–436.

[17] Erman, L. D.—Lesser, V. R.: A Multi-Level Organization for Problem Solving Using Many Diverse, Cooperating Sources of Knowledge. Proceedings of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Russia, 1975, pp. 483–490.

[18] Ziebarth, S.—Malzahn, N.—Hoppe, H. U.: Matchballs – A Multi-Agent-System for Ontology-Based Collabortaive Learning Games. In: Herskovic, V., Hoppe, H. U., Jansen, M., Ziegler, J. (Eds.): 18$^{\text{th}}$ International Conference on Collaboration and Technology (CRIWG 2012), Raesfeld, Germany, Springer, 2012, pp. 208–222.

[19] Miles, A.—Bechofer, S.: SKOS Simple Knowledge Organization System Reference. 2009.

[20] Ziebarth, S.—Malzahn, N.—Hoppe, H. U.: Matchballs – A Casual Game for Learning and Domain Ontology Enrichment. International Conference of the Learning Sciences (ICLS 2012), Sydney, Australia, 2012.

[21] MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[22] Novak, J. D.—Canás, A. J.: The Theory Underlying Concept Maps and How to Construct Them. Technical Report IHMC CmapTools, Florida Institute for Human and Machine Cognition, 2006.

[23] Esiobu, G.—Soyibo, K.: Effects of Concept and Vee Mapping Under Three Learning Modes on Students' Cognitive Achievement in Ecology and Genetics. Journal of Research in Science Teaching, Vol. 32, 1994, No. 9, pp. 971–995.

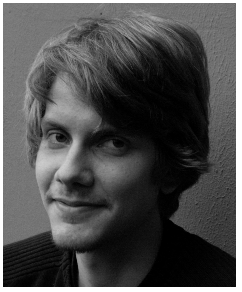
[24] Chung, G. K. W. K.—O'Neil, H. F. Jr.—Herl, H. E.: The Use of Computer-Based Collaborative Knowledge Mapping to Measure Team Processes and Team Outcomes. Computers in Human Behavior, Vol. 15, 1999, No. 3–4, pp. 463–493.

[25] Chinn, C. A.—O'Donnell, A. M.—Jinks, T. S.: The Structure of Discourse in Collaborative Learning. Journal of Experimental Education, Vol. 69, 2000, No. 1, pp. 77–97.

[26] Chiu, C. H.—Huang, C. C.—Chang, W. T.: The Evaluation and Influence of Interaction in Network Supported Collaborative Concept Mapping. Computers & Education, Vol. 34, 2000, No. 1, pp. 17–25.

Nils Malzahn is a doctoral student at the Collide Research Group. He is interested in ontology-based learning applications and learning of ontologies from various sources as well as distributed multi-agent software architectures. As of late he is working as a senior researcher for the Rhine-Ruhr Institute for Applied System Innovation e.V. in Duisburg, Germany.

Sabrina Ziebarth is a postdoctoral researcher at the Collide Research Group. She is especially interested in ontology learning and ontology-based learning applications.



Dominik Kloke was a student assistant at the Collide Research Group. He developed the Knowledge Inspector and continues his studies of computer science at the HU Berlin, Germany.

H. Ulrich Hoppe holds a full professorship for cooperative and learning support systems in the Department of Computer Science and Applied Cognitive Science at the University of Duisburg-Essen, Germany. He is the founder of the research group Collide (Collaborative Learning in Intelligent Distributed Environments). His current research interests include the analysis, modelling, and intelligent support of interactive and collaborative learning as well as social network analysis and community support.