

Knowledge-based Learning Content Generation in the STUDYBATTLES Environment

Amal Shehadeh, Alexander Felfernig, Martin Stettinger

Institute for Software Technology

Graz University of Technology

Graz, Austria

{ashehade,alexander.felfernig,martin.stettinger}@ist.tugraz.at

Michael Jeran, Stefan Reiterer

SelectionArts GmbH

<http://www.selectionarts.com>

Graz, Austria

{m.jeran,s.reiterer}@selectionarts.com

Abstract— *E-learning environments provide an orthogonal approach to transfer relevant knowledge. For example, sales representatives can improve their sales knowledge more independently from related courses offered. Major challenges for successfully establishing e-learning technologies in a company are to develop learning content in an efficient fashion, to recommend only relevant content to system users, and to motivate them to utilize the learning environment in a sustainable fashion. In this paper, we present the gamification-based e-learning environment STUDYBATTLES. We provide an overview of STUDYBATTLES functionalities including content creation, gamification techniques, learning performance analysis, and automated question generation. We show how STUDYBATTLES is and can be utilized for different learning purposes in academic and professional environments. In addition, we introduce an approach to automatically generate product and sales domain learning content from recommender knowledge bases to be exploited in STUDYBATTLES. Finally, we report the results of an initial qualitative study related to the applicability of STUDYBATTLES in different domains, the potential improvements that STUDYBATTLES can achieve, and additional functionalities that should be integrated.*

Keywords— *gamification-based e-learning; automated question generation; knowledge-based recommender systems; constraint satisfaction problem; knowledge acquisition*

1 INTRODUCTION

E-learning [13] provides an orthogonal approach to knowledge transfer in sectors such as companies, public organizations, universities, and schools. Companies apply e-learning environments for establishing a *corporate memory* that can be (and often has to be) used by employees to improve their process-relevant knowledge (e.g., marketing, sales, and product-specific knowledge). Similarly, in the public sector, employees establish process-relevant knowledge often on the basis of e-learning environments. Although it's also applicable to transfer process-relevant knowledge, employees at universities and schools apply e-learning environments primarily for the purpose of improving the quality of teaching and to reduce related time efforts.

Improvements triggered by the application of e-learning environments are manifold. E-learning environments make learning material widely accessible to users and thus create *flexibility with regard to the time of learning and training*. For organizations, it becomes *easier to understand strengths and weaknesses of employees with regard to organization-related knowledge*.

An entry barrier to the usage of e-learning systems, especially in companies, is the absence of relevant content, i.e., additional investments are needed before being able to ramp up the system. An approach to tackle this challenge is to provide *crowd sourcing* concepts which allow expert users on their own to provide the relevant content. This approach avoids knowledge acquisition bottlenecks since knowledge acquisition tasks become decentralized.

Content creation can be made even more efficient by *automatically generating learning content from different types of knowledge*

sources [11]. In the remainder of this section we will shortly analyze existing approaches to content generation and then focus on the content generation approach implemented in STUDYBATTLES.

Agarwal et al. [2] introduce an automatic question generation system that can generate gap-fill questions from a document. Gap-fill questions are fill-in-the-blank questions with multiple choices provided (one correct answer and three distractors). The system selects the most informative sentences of the chapters and generates gap-fill questions thereof by first blanking keys from the sentences and then determining the distractors for these keys. Syntactic and lexical features are used in this process without relying on any external resource apart from the information in the document.

Hussein et al. [8] propose a system which generates questions from a document by selecting one sentence at a time, extracts sections of the source sentence, then applies transformation rules for constructing a question. The proposed system is based on the OpenNLP open source statistical parser to generate the questions.

Gütl et al. [7] introduce a concept and prototype implementation of the Enhanced Automatic Question Creator (EAQC) which extracts most important concepts out of textual learning content and creates single choice, multiple-choice, completion exercises, and open ended questions on the basis of these concepts. Their approach combines statistical, structural, and semantic methods of natural language processing as well as a rule-based solution for concept extraction. EAQC is designed to deal with multilingual learning material.

Afzal [1] introduces an approach to automatically generate multiple-choice questions from text. They extract semantic relations and then automatically generate questions using these semantic relations. They evaluated the whole system, for example, in terms of question understandability, usefulness of semantic relations, and overall multiple choice question usability and found out that the generated questions are considered as relevant.

Alsubait et al. [3, 14] present an approach to automatically generate multiple-choice questions from OWL ontologies. They present an empirical evaluation of an ontology-based question generation approach and examine the feasibility of applying ontology-based question generation by educators with no prior experience in ontology building. They found out that this approach can result in a reasonable number of educationally useful questions.

In this paper, we introduce an approach that has commonalities with the question generation approach introduced in [3, 14]. In the STUDYBATTLES environment, a *recommender knowledge base* can be used for question generation where questions do not only refer to the *solution space* defined by the knowledge base but also to situations where *no solution can be found* for a given set of customer requirements. Compared to the approach presented in [3, 14], our question generation approach is based on a recommender knowledge base represented as a Constraint Satisfaction Problem (CSP) [4] which is exploited for automated question generation.

Our contributions in this paper are the following. First, we provide an overview of the STUDYBATTLES e-learning environment. Second, we introduce a set of new techniques that can be applied for *automated question generation out of constraint-based recommendation knowledge bases*. Finally, we report initial results of a qualitative

study related to the applicability and effectiveness of STUDYBATTLES.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the major functionalities of the STUDYBATTLES e-learning environment. In Section 3, we present a working example from the domain of automotive services (we selected this domain as a working example because it is easy to understand). In Section 4, we introduce an approach to the automated question generation from recommender knowledge bases – these questions can then be exploited by STUDYBATTLES. An empirical evaluation of the functionalities provided by STUDYBATTLES is presented in Section 5. The paper is concluded with a discussion of issues for future work in Section 6.

2 STUDYBATTLES ENVIRONMENT

STUDYBATTLES is an e-learning environment that has been developed within the PEOPLEVIEWS research project¹ at Graz University of Technology. The STUDYBATTLES start screen is depicted in Fig. 1 which includes a short system description, a list of subscriptions to learning applications, and further learning applications to subscribe to. STUDYBATTLES mobile clients are available as iOS, Android, and HTML-5 web client. The system itself is available as global platform² and as an in-house server solution for individual enterprises. STUDYBATTLES learners can join communities and subscribe to learning applications where they can practice exercises, add content, and compete against each other in quiz-based duels. STUDYBATTLES supports content categorization in the learning applications, allowing for better structuring and easy information access for learners.

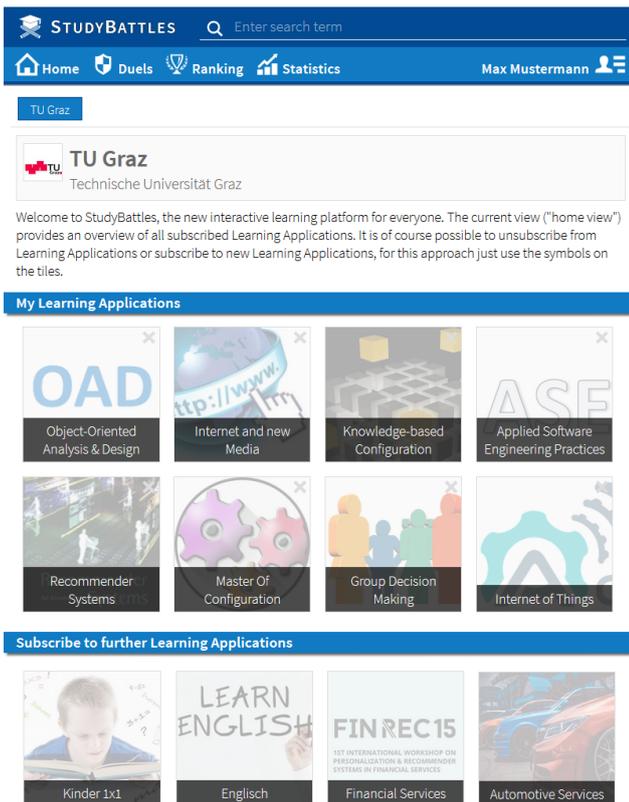


Figure 1. STUDYBATTLES start screen (HTML-5 version) consisting of a set of learning applications that can be selected by the user.

STUDYBATTLES deployments. The system is already deployed

¹ The work presented in this paper has been conducted within the scope of the PEOPLEVIEWS project funded by the Austrian Research Promotion Agency (843492).

² www.studybattles.com

and applied at one municipality and two universities in Austria. At the mentioned universities, STUDYBATTLES is applied in three Software Engineering courses and in two Artificial Intelligence related courses. The goal of the STUDYBATTLES instance deployed at the mentioned Austrian municipality is to *increase employees knowledge with regard to security-related topics* and to transfer application-oriented knowledge related to *a new accounting system*. Currently, STUDYBATTLES is also deployed for one of the largest financial service providers in Austria. The major goal of this deployment is to support sales representatives in their learning processes related to *product knowledge and sales practices*. STUDYBATTLES users from these domains have been interviewed within the scope of our qualitative study reported in Section 5. Furthermore, STUDYBATTLES is currently used within the scope of STUDYBATTLES:HELP project³ to help refugees to quickly learn important and needed topics related to their integration into the Austrian society.

Learning and training. After subscribing to a STUDYBATTLES learning application, the user can select categories and questions to answer. After answering a question, the user receives an immediate feedback on the correctness of his or her answer. If the answer is incorrect, a related explanation is provided to the user (in case it has been included during the definition phase of the question).

Content Creation and Question types. STUDYBATTLES follows the concept of *crowd sourcing* where users can enter questions and expert users can evaluate the quality of the questions. The status of a *domain expert* is reached if a certain threshold of correctly answered questions is passed. Users are allowed also to add additional content in terms of text, documents, pictures, and movies which serve as a basis for answering questions.

STUDYBATTLES supports different types of questions, examples are depicted in the Figures 2– 5. Fig. 2 depicts a simple example of a *multiple-choice question* – the question is related to the relationship between customer requirements and corresponding products (cars). The abbreviations used in Fig. 2 represent customer requirements: *pu* = *purpose of use*, *cat* = *category*, and *fc* = *fuel consumption*.

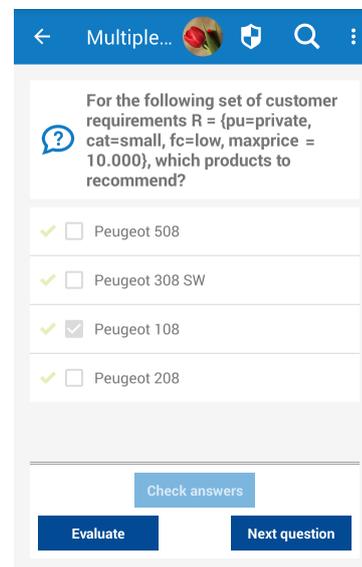


Figure 2. STUDYBATTLES: representation of multiple-choice questions (Android version). The green check marks represent the correctness feedback (*pu* = *purpose of use*, *cat* = *category*, and *fc* = *fuel consumption*).

Fig. 3 depicts an example of an *association task* where terms on the right-hand side have to be associated with the corresponding terms on the left-hand side. The corresponding HTML-based definition interface is depicted in Figure 6. Association tasks can be exploited, for example, for (1) asking questions regarding compati-

³ helpers.studybattles.com

bility relationships between customer requirements and products. (2) asking questions about the incompatibility of specific customer requirements. In the example of Fig. 4 users are requested to associate inconsistent customer requirements on the left and right hand side.

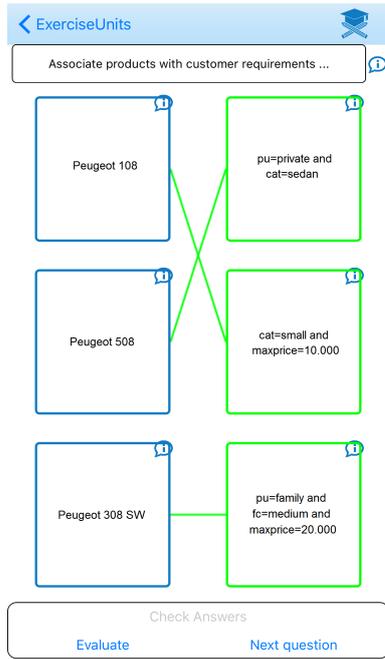


Figure 3. STUDYBATTLES: representation of an associations task (iOS version) – products on the left have to be associated with corresponding compatible requirements on the right.

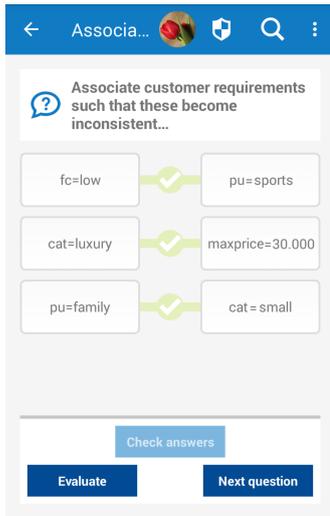


Figure 4. STUDYBATTLES: Association task related to the association of inconsistent requirements (Android version).

In Fig. 5 a question is posed to educate users with regard to repair knowledge (which situations lead to an empty set of solutions and how can a user recover from such situations).

Gamification. STUDYBATTLES supports several gamification techniques such as points, levels, ranking, badges, feedback, and duels. Users can trigger quiz-based duels and then are assigned randomly to opponents. Alternatively, users can play against the best performing user in a certain learning application – the ranking of a user is visible “locally”, i.e., a user is only able to see other users directly ranked before and after him/her. STUDYBATTLES points can be collected if a user is the winner of a duel. There are other opportunities to collect STUDYBATTLES points, for example, when answering a question correctly or adding a new question to the system.

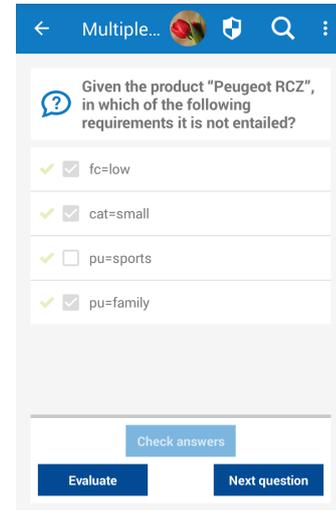


Figure 5. STUDYBATTLES: representation of inconsistency-related knowledge in terms of a multiple-choice question.

The higher the estimated complexity of the question, the higher is the number of received STUDYBATTLES points. The complexity of a question can be evaluated directly after having answered the question (see the Evaluate button, for example, in Fig. 5)

Analysis of learning performance. STUDYBATTLES supports different types of statistics (for users and administrators) that help to analyze the strengths and weaknesses of the user community and to establish needed counter measures (e.g., improving/adapting some parts of the learning material). The statistics are provided on the Learning application, categories, and user levels.

3 A SIMPLE AUTOMOTIVE SERVICE ADVISOR

Knowledge-based recommender systems [9] are conversational systems where users are enabled to specify their preferences and the system proposes a corresponding solution. Constraint-based recommender systems [4] are a specific type of knowledge-based recommenders where recommendation knowledge is defined in terms of a constraint satisfaction problem (CSP) – see the following definition.

Definition (Recommendation Task). A recommendation task can be defined as a CSP $(V, D, C, REQ, PROD)$ where V is a set of variables describing potential customer requirements and product properties, D represents domain definitions for the variables, C is a set of constraints, REQ represents a set of customer requirements, and $PROD$ is the products catalog.

Definition (Recommendation). A recommendation (solution) for a given recommendation task $(V, D, C, REQ, PROD)$ is a complete set A of variable assignments $v_i = a$ to the variables $v_i \in V (v_i = a \rightarrow a \in \text{domain}(v_i))$ with $\text{consistent}(A \cup C \cup REQ \cup PROD)$.

Consequently, solutions determined for recommendation tasks can be considered as candidate recommendations for a customer (user). Alternative solutions can be further ranked according to their utility for the customer – for details on how to rank such candidate sets we refer to [9]. We now introduce a simple *Automotive Service Advisor knowledge base* that will be used as working example throughout the paper. The variables in V are *purpose of use* (pu), *category* (cat), *fuel consumption* (fc), *maxprice*, *price*, and *name* which represents the name of the car.

- $V = \{pu, cat, fc, maxprice, price, name\}$
- $D = \{\text{domain}(pu) = \{\text{family, private, sports}\}, \text{domain}(cat) = \{\text{sedan, coupe, luxury, MPV}^4, \text{small}\}, \text{domain}(fc) =$

⁴ Multi-Purpose Vehicle (MPV): a car classification used in Europe to describe small family cars.

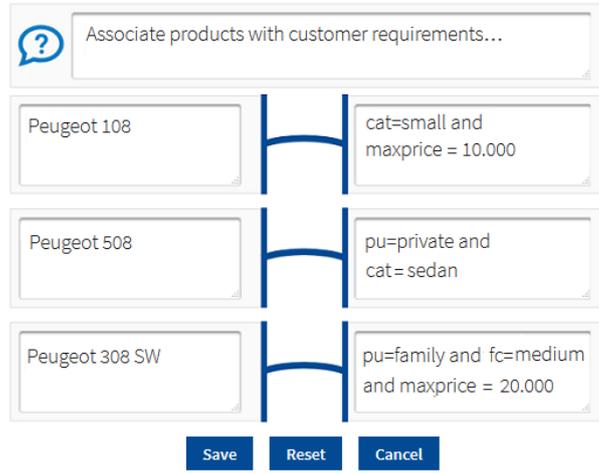


Figure 6. STUDYBATTLES: definition of an association task (specification of correct pairs) in the HTML-5 version.

{low, medium, high}, $\text{domain}(maxprice) = \{10.000, 20.000, 30.000, 40.000, 50.000\}$, $\text{domain}(price) = [9.000-50.000]$, $\text{domain}(name) = \{\text{Peugeot 5008, Peugeot 308 SW, Peugeot 108, Peugeot 208, Peugeot 508, Peugeot RCZ, Peugeot 308 GTi, Peugeot 4008 380THP}\}$

- $C = \{c_1 : price \leq maxprice, c_2 : cat = small \rightarrow fc = low, c_3 : pu = family \rightarrow cat \notin \{small, coupe\}, c_4 : cat = luxury \rightarrow fc = high, c_5 : pu = private \vee pu = sports \rightarrow cat \neq MPV, c_6 : cat = small \rightarrow price < 12.000, c_7 : \neg(pu = sports \wedge fc = low), c_8 : cat = luxury \vee pu = sports \rightarrow maxprice \in \{40.000, 50.000\}\}$
- $REQ = \{r_1 : pu = private, r_2 : fc = low, r_3 : maxprice = 10.000\}$
- $PROD$ is the product catalog which represents the constraints responsible for restricting the possible instantiations of the variables in V . A simple example of a product catalog for our automotive recommender is shown in Table 1.

Table 1. product catalog ($pu = purpose of use$, $cat = category$, and $fc = fuel consumption$).

name	pu	cat	fc	price
Peugeot 5008	family	MPV	medium	27.030
Peugeot 308 SW	family	MPV	medium	18.213
Peugeot 108	private	small	low	9.399
Peugeot 208	private	small	low	11.971
Peugeot 508	private	sedan	high	28.503
Peugeot RCZ	sports	coupe	high	49.990
Peugeot 308 GTi	sports	sedan	high	36.143
Peugeot 4008 380THP	family	luxury	high	37.543

A recommendation for the given example recommendation task is the set of variable assignments $A = \{pu = private, cat = small, fc = low, price = 9.399, name = Peugeot108\}$. In this situation, the given set of customer requirements defined in REQ are consistent with the constraints in C . However, even a slight change in the specification of REQ can lead to a situation where the customer requirements become inconsistent with the constraints in C . For example, if we define $REQ = \{r_1 : pu = family, r_2 : fc = low, r_3 : maxprice = 10.000\}$, no solution can be found.

In situations where no solution can be found for a given set of requirements, concepts of model-based diagnosis [12] can help to identify a minimal set of requirements that have to be adapted such that a solution can be identified. For the identification of such minimal changes, we introduce the definition of a diagnosis task and a corresponding diagnosis (see the following definitions).

Definition (Diagnosis Task). A diagnosis task is defined as a tuple $(C, REQ, PROD)$ where C is a set of constraints, REQ is a set of customer requirements, $PROD$ is the product catalog, and $REQ \cup C \cup PROD$ is inconsistent.

Definition (Diagnosis). A set $\Delta \subseteq REQ$ for a given diagnosis task $(C, REQ, PROD)$ is a diagnosis if $REQ - \Delta \cup C \cup PROD$ is consistent, i.e., Δ is a set of requirements (constraints) that have to be deleted from REQ such that the remaining customer requirements are consistent with $C \cup PROD$. Furthermore, Δ is minimal if there does not exist a set Δ' with $\Delta' \subset \Delta$.

The basic algorithm for determining minimal diagnoses is introduced in [12]. For further algorithms specifically designed to be applied in interactive settings we refer to [5]. All these algorithms are complete, i.e., they are able to determine the complete set of minimal diagnoses on the basis of the concepts of hitting set directed acyclic graphs introduced in [12] – this is an important property that can also be exploited in the context of question generation (see the following section).

Types of sales knowledge. Knowledge-based recommender systems include sales knowledge in different forms. First, given a set of customer requirements, a recommender can determine items that can be recommended (*filter knowledge*). Second, given an item, a recommender can determine customer requirements that are consistent with the item (*product knowledge*). Third, in situations where no solution can be identified for a given set of customer requirements, diagnosis can determine the needed minimal changes to help the user out of the *no solution could be found dilemma* (*analysis knowledge*). Fourth, given an item, a recommender can determine a set of customer requirements which are inconsistent with the item (*inconsistency knowledge*). In the following, we discuss how recommendation task definitions can be exploited for the generation of questions for an e-learning environment that is used, for example, for the education of sales representatives.

4 GENERATING QUESTIONS FROM RECOMMENDATION TASK DEFINITIONS

Questions and corresponding answers for the STUDYBATTLES environment can be automatically generated from the constraints contained in a recommendation task definition. On the basis of our working example, we now introduce an approach to automatically generate questions and related answers for the four aforementioned types of sales knowledge. The generated questions and answers can be imported into STUDYBATTLES and then be used for training and supporting sales representatives. The overall goal of these questions is to increase the personal level of sales knowledge and, as a consequence, to make advisory services more efficient.

4.1 Filter knowledge related question generation

The underlying task is to figure out which items fit a given set of predefined customer requirements (REQ). More formally, filter knowl-

edge related questions can be generated on the basis of a recommendation task definition ($V, D, C, REQ, PROD$). On the basis of such definition, a constraint solver (recommender) is able to determine all possible instantiations of customer requirements and corresponding items. Each set of customer requirements can then be the basis of the question, the corresponding items represent the correct answer(s).

Example. Given the recommendation task definition of our working example (Automotive Services Advisor), a set of customer requirements could be $REQ = \{r_1 : cat = MPV, r_2 : fc = medium, r_3 : maxprice = 20.000\}$, the set of corresponding correct answers is $\{name = Peugeot308SW\}$. The corresponding question that would be posed in the STUDYBATTLES environment is: **Given the following customer requirements ..., which item(s) would you recommend?**

4.2 Product knowledge related question generation

The underlying task is to figure out which sets of customer requirements fit a given item. In this context, for each possible item a constraint solver can return the complete collection of sets of customer requirements which are consistent with this item.

Example. Given the recommendation task definition of our working example, the item selected could be $name = Peugeot308SW$. The corresponding collection of consistent customer requirements is $\{R_1 : \{r_1 : pu = family, r_2 : cat = MPV, r_3 : fc = medium, r_4 : maxprice = 20.000\}, R_2 : \{r_1 : pu = family, r_2 : fc = medium, r_3 : maxprice = 20.000\}, R_3 : \{r_1 : pu = family, r_2 : cat = MPV, r_3 : maxprice = 20.000\}, R_4 : \{r_1 : cat = MPV, r_2 : fc = medium, r_3 : maxprice = 20.000\}$, i.e., in this example there exist 4 sets R_i of customer requirements that are consistent with the selected item $Peugeot308SW$. The corresponding question that would be posed in the STUDYBATTLES environment is: **Given the following item ..., which sets of customer requirements are consistent with that item?**

4.3 Analysis knowledge related question generation

The underlying task is to figure out which minimal sets of customer requirements have to be adapted such that a solution can be found. More formally, analysis knowledge related questions can be generated on the basis of a diagnosis task definition ($C, REQ, PROD$). The diagnosis task definition can be used for question generation, the related answers are represented by the determined diagnoses Δ_i .

Example. Given the recommendation task definition of our working example with an adapted set of customer requirements $REQ = \{r_1 : pu = sports, r_2 : cat = coupe, r_3 : fc = low\}$, which is inconsistent with $(C \cup PROD)$, the corresponding alternative minimal sets of customer requirements (diagnoses Δ_i) that have to be deleted from REQ such that a solution can be identified, are the following: $\{\Delta_1 = \{r_1, r_2\}, \Delta_2 = \{r_3\}\}$. For example, deleting (or adapting) the requirement r_3 restores consistency, i.e., allows the calculation of a recommendation. The corresponding question that would be posed in STUDYBATTLES is: **Given the following recommendation task definition ... which one is a minimal set of requirements that have to be deleted from REQ such that a recommendation can be identified?**

4.4 Inconsistency knowledge related question generation

The underlying task is to figure out which sets of customer requirements trigger an inconsistency with a pre-selected item. More formally, inconsistency knowledge related questions can be generated on the basis of a recommendation task definition ($V, D, C, REQ, PROD$) where all combinations of customer requirements have to be determined that never include the pre-selected item.

Example. In our working example, we could pre-select the item $name = PeugeotRCZ$. Combinations of customer requirements that do not entail the item $name = PeugeotRCZ$ are all possible combinations with the exception of the following combination: $\{pu = sports, cat = coupe, fc = high, maxprice = 50.000\}$. The corresponding question that would be posed in STUDYBATTLES is: **Given the following item ... which combination of customer requirements does not entail this item?**

5 STUDYBATTLES EVALUATION

In order to evaluate the applicability and effectiveness of the STUDYBATTLES environment, we conducted a qualitative study with experts $N=20$ from different domains (financial services, public administration, telecommunications, and universities). In this study, domain experts provided feedback related to major potential improvements that come along with the application of STUDYBATTLES, the possible domains of application, and what additional functionalities should be integrated into the system⁵.

Potential improvements due to the application of STUDYBATTLES. Within the scope of this study, the potential improvements that have been pointed out are the following: improved knowledge retention in organizations (See Fig. 7), improved knowledge sharing between users on the basis of community-based (crowd-sourced) knowledge acquisition processes, increased motivation to learn, improved skills, increased fun and interest in the topic, increased competition level between users (e.g., sales representatives), improved quality of service with regard to customers (e.g., due to the already discussed question generation mechanisms - see Fig. 8, reduced time efforts in the learning process and services (e.g., sales representatives and advisory services in the automotive domain - see Fig. 9), and enhanced possibilities of community knowledge analysis which provide a basis for a more fine-grained adaptation of learning material where needed.

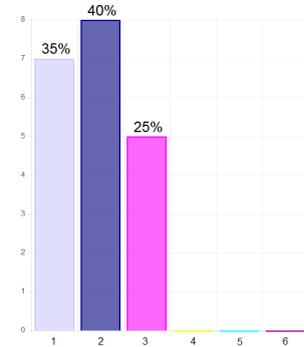


Figure 7. STATISTICS: Effectiveness of StudyBattles - Improved knowledge retention in the organization: (1) Excellent, (2) Very Good, (3) Good, (4) Neutral, (5) Fair, (6) Poor

Application scenarios. In addition to the application domains where STUDYBATTLES has already been deployed, the participants of the study mentioned the following additional application scenarios where STUDYBATTLES could be applied: in the *health domain*, doctors have additional means to keep their knowledge up-to-date and also to get confronted with new health-related knowledge in a more systematic fashion. *University personnel* can apply STUDYBATTLES not only for teaching purposes but also for providing e-learning content to new employees (e.g., PhD students in their first semester who are in need of learning about the internal processes and rules). *Manufacturers* (e.g., in the automotive domain) can apply the system to improve the production processes and sales related knowledge of employees. Further mentioned application domains are *schools*, *societies* (intuitive explanation of rules for new members), and a *global*

⁵ Statistics that illustrate the results cannot be displayed all in the paper for space reasons.

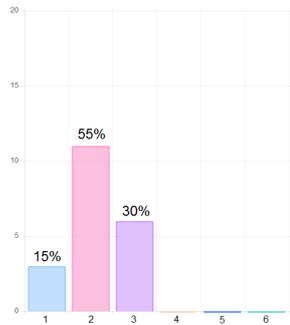


Figure 8. STATISTICS: Effectiveness of StudyBattles - Improved quality of services.

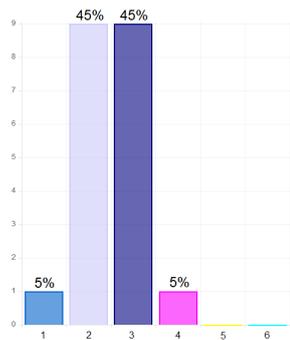


Figure 9. STATISTICS: Effectiveness of StudyBattles - Reduced time efforts in services.

platform for knowledge provision and consumption that can be applied by everyone.

Additional functionalities. Another focus of the qualitative study was to figure out additional functionalities that should be integrated into STUDYBATTLES. Functionalities mentioned by the participants are the following: with the generated questions (see Section 4), a more fine-grained determination of STUDYBATTLES points could be achieved, for example, by also taking into account the prediction quality of sales representatives (in one scenario, questions are related to the filter knowledge of sales representatives – the better the predictions are, the more corresponding points should be assigned to the user). *Social network* functionalities should be included since these provide an additional means to achieve a more frequent usage of STUDYBATTLES, for example, due to personalized status messages. Beside the already included personalized ranking of questions, additional recommendation functionalities should be included, for example, the recommendation of opponents in duels, the recommendation of potential friends in the mentioned social network, and the recommendation of additional learning content. Finally, participants of the study proposed the inclusion of intelligent user interface elements that help to increase the engagement in STUDYBATTLES, for example, a kind of traffic light feedback interface could motivate more frequent interaction with the system (see also [10]).

6 CONCLUSIONS

In this paper, we provided an overview of the e-learning environment STUDYBATTLES and how it can be used as a complementary approach to transfer knowledge related to product and sales practices.

Artificial Intelligence concepts currently included in the STUDYBATTLES environment are basic recommendation mechanisms related to the selection of questions to be shown to users, and mechanisms that support the automated generation of questions and related correct answers. Future work will focus on an extension of the provided recommendation functionalities, for example, by recommending users for duels, recommending learning applications, recommending content in a collaborative fashion, and recommending

users for an inclusion in specific learning teams. Furthermore, we will extend our approach to learning content generation, for example, by allowing to control the degree of complexity of the content. In this context, we will also analyze potential synergies with existing approaches to test case generation in software engineering.

Especially in the context of educating sales representatives, automated question generation becomes a key functionality, since it reduces the overheads of manual content generation and management, which is often the task of only a small group of persons. In future versions of STUDYBATTLES, additional question types will be included. For example, we will provide mechanisms that allow to generate not only questions related to diagnoses (analysis knowledge) but also to related repair actions (i.e., changes in the requirements that lead to the identification of at least one solution). In addition, we will use approaches to reduce the number of calculated diagnoses, such as *minimal cardinality diagnoses* [12][6], to identify the diagnoses that are more relevant to users (especially in the case of large sets of diagnosis alternatives). Furthermore, diagnosis personalization techniques [6] will be exploited to improve the diagnosis prediction quality, by integrating recommendation approaches such as similarity-based, utility-based, probability-based, and ensemble-based with standard model-based diagnosis [12].

REFERENCES

- [1] N. Afzal, 'Automatic generation of multiple choice questions using surface-based semantic relations', *International Journal of Computational Linguistics (IJCL)*, **6**, 26–44, (2015).
- [2] M. Agarwal and M. Prashanth, 'Automatic gap-fill question generation from text books', in *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, IUNLPBEA '11*, pp. 56–64. Association for Computational Linguistics, (2011).
- [3] Tahani Alsubait, Bijan Parsia, and Uli Sattler, 'Generating multiple choice questions from ontologies: lessons learnt', in *Proceedings of the 11th International Workshop on OWL: Experiences and Directions (OWLED 2014)*, pp. 73–84. CEUR-WS.org, (2014).
- [4] A. Felfernig and R. Burke, 'Constraint-based recommender systems: technologies and research issues', in *Proceedings of the 10th International Conference on Electronic Commerce, ICEC '08*, pp. 3:1–3:10. ACM, (2008).
- [5] A. Felfernig, M. Schubert, and C. Zehentner, 'An efficient diagnosis algorithm for inconsistent constraint sets', *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, **26**(1), 53–62, (2012).
- [6] Alexander Felfernig, Monika Schubert, and Stefan Reiterer, 'Personalized diagnosis for over-constrained problems', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13*, pp. 1990–1996. AAAI Press, (2013).
- [7] C. Gütl, K. Lankmayr, J. Weinhofer, and M. Höfler, 'Enhanced automatic question creator - EAQC: concept, development and evaluation of an automatic test item creation tool to foster modern e-education', *Electronic Journal of e-Learning*, **9**, (2011).
- [8] H. Hussein, M. Elmogy, and S. Guirguis, 'Automatic English question generation system based on template driven scheme', *IJCSI International Journal of Computer Science*, **11**(1), (2014).
- [9] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender systems: an introduction*, Cambridge University Press, 1st edn., 2010.
- [10] G. Ninaus, A. Felfernig, M. Stettinger, S. Reiterer, G. Leitner, L. Weninger, and W. Schanil, 'INTELLIREQ: intelligent techniques for software requirements engineering', in *ECAI 2014 - 21st European Conference on Artificial Intelligence*, pp. 1161–1166. IOS Press, (2014).
- [11] S. Rakangor and Y. Ghodasara, 'Literature review of automatic question generation systems', *International Journal of Scientific and Research Publications*, **5**, (2015).
- [12] R. Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence Journal*, **32**(1), 57–95, (April 1987).
- [13] L. Shen and R. Shen, *Advances in web-based learning – ICWL 2004: Third International Conference Proceedings*, chapter Learning content recommendation service based on simple sequencing specification, 363–370, Springer Berlin Heidelberg, 2004.
- [14] A. Tahani, B. Parsia, and U. Sattler, *Knowledge engineering and knowledge management: EKAW 2014. Revised Selected Papers.*, chapter Generating multiple choice questions from ontologies: how far can we go?, 66–79, Springer International Publishing, 2015.