# StudyBattles: A Learning Environment for Knowledge-based Configuration

**Alexander Felfernig**[1] and **Amal Shehadeh**[1] and **Michael Jeran**[1] and **Christian Gütl**[2] and **Trang Tran**[1] and **Müslüm Atas**[1] and **Seda Polat Erdeniz**[1] and **Martin Stettinger**[1] and **Arda Akcay**[1] and **Stefan Reiterer**[3]

**Abstract.** E-learning is a complementary channel for learners to acquire relevant knowledge. On the basis of an example e-learning environment (StudyBattles) we show how configuration-related knowledge can be transferred to end-users (e.g., sales representatives) as well as to knowledge engineers. In addition, we discuss an approach to automatically generate product domain as well as engineering learning content to be used in StudyBattles. Finally, we report the results of an initial qualitative study on the applicability of StudyBattles.

## 1 Introduction

Knowledge-based configuration is one of the most successfully applied Artificial Intelligence technologies [1, 4, 9, 15]. Configuration systems improve business processes in various dimensions such as *reduced error rates* and *time efforts* in product offering and *reduced costs* of error management in follow-up production processes. Despite the successful application of configuration technologies, there are still issues related to the transfer of configuration related knowledge to employees.

In dialogs with customers, sales representatives should not only rely on solutions and related explanations provided by the configuration environment but should also have the needed domain knowledge. Furthermore, engineers and domain experts engaged in knowledge engineering processes should have the needed technical foundations and be aware of engineering practices to minimize overheads in knowledge engineering processes. Finally, domain experts in charge of documenting configuration knowledge should be aware of standards on how to document knowledge in such a way that knowledge engineers can formalize this knowledge easily. The goal of this paper is to show how e-learning technologies [16] can be applied as a means (in addition to traditional training programs such as sales force training or trainings related to knowledge acquisition and maintenance) to support the mentioned knowledge transfer.

E-learning systems are often applied for creating a *corporate memory* that is exploited to improve process-relevant knowledge of employees (e.g., sales, marketing, and product management). Improvements triggered by the application of e-learning technologies are manyfold. They reach from the *increased accessibility* of learning contents (users are much more flexible with regard to the time of learning and training), *increased opportunities to analyze the strengths and weaknesses of employees with regard to organization-relevant knowledge*, and *increased consumption frequency of learning content* due to the application of different types of motivation mechanisms (e.g., gamification and persuasion [7]).

In this paper we focus on two configuration-related types of knowledge. *First*, we show how *sales-relevant configuration knowledge* can be represented in an e-learning environment. Examples of such knowledge types are *product knowledge* (e.g., for a specific set of customer requirements, which configurations should be recommended) and *analysis knowledge* (e.g., if no solution (configuration) can be identified for a given set of customer requirements, which alternatives should be proposed to the customer in order to maximize the probability that the customer will accept the offer).

*Second*, especially less experienced knowledge engineers and domain experts should be educated with regard to *best practices in knowledge acquisition and maintenance*. Examples of such knowledge types are *documentation knowledge* (e.g., in which way should incompatibilities between components types be documented on a textual level) and *knowledge representation knowledge* (e.g., in which context one should use compatibilities or incompatibilities to express allowed combinations of component types).

The existing demand for complementary means of transferring configuration-relevant knowledge to employees has already been identified in earlier works. For example, Felfernig et al. [5] introduce a gamification-based approach to learning the major technical concepts of knowledge-based configuration and model-based diagnosis [13] – initial results of their studies show that the learning success of students can be increased. Compared to this approach, StudyBattles does not only support the dissemination of technical product configuration knowledge but also allows to include product domain knowledge into e-learning processes. Furthermore, StudyBattles includes gamification concepts which are implemented as *duels* where different users can play against each other in the context of a specific pre-selected learning application.

Felfernig et al. [6] analyze existing misconceptions of knowledge engineers when interpreting textual domain descriptions and recommend different measures that can help to reduce efforts related to configuration knowledge acquisition and maintenance. Finally, an analysis of the cognitive complexity of different types of knowledge formalizations is presented in [14] – the authors show that different types of representing logical implications can lead to significantly different outcomes in terms of knowledge understandability. Fur-

[1] Institute for Software Technology, Graz University of Technology, Austria, email: alexander.felfernig@ist.tugraz.at. ashehade@ist.tugraz.at. mjeran@ist.tugraz.at.

[2] Institute for Information Systems and Computer Media, Graz University of Technology, Austria and Curtin University, Western Australia, email: Christian.Guetl@iicm.tugraz.at.

[3] SelectionArts, Austria, email: s.reiterer@selectionarts.com.

thermore, different approaches to structure constraints can also have an impact on the underlying degree of understandability. Results of these studies have been integrated into a STUDYBATTLES learning application (see Figure 1).

Besides exploiting user communities, e-learning content creation can be made more efficient by *automated generation generation mechanisms* – see, for example, [8, 12]. In this paper we show how configuration-related learning content (questions and related answers) can be automatically generated from a given configuration knowledge base. In addition to existing approaches to question generation from formal representations, we do not only focus on questions that refer to the set of possible solutions (configurations). We also show how questions can be generated that refer to *inconsistent situations* (e.g., no solution can be identified for a given set of customer requirements or the knowledge base becomes inconsistent with a given set of test cases) and to *qualitative properties of knowledge bases* (e.g., redundancies and further well-formedness properties).

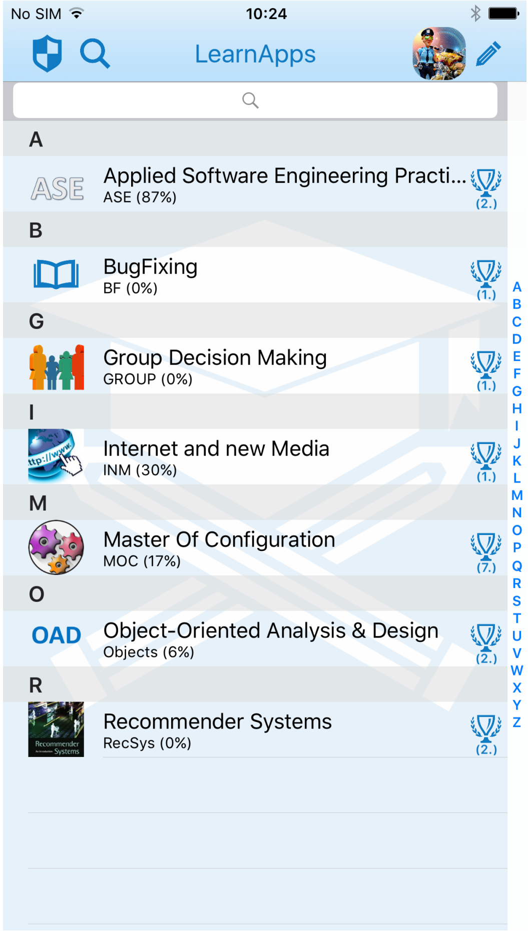
Our contributions in this paper are the following. First, we provide an overview of the STUDYBATTLES e-learning environment and show how the mentioned types of configuration-related knowledge can be represented in the system. Second, we show how configuration knowledge bases can be exploited to automatically generate e-learning content. Finally, we report initial results of a qualitative study related to the applicability of STUDYBATTLES.

The remainder of this paper is organized as follows. In Section 2, we provide a short overview of the different functionalities provided in STUDYBATTLES. As a basis for introducing a question generation approach, we define a configuration knowledge base from the domain of financial services that serves as a working example throughout this paper (Section 3). In Section 4 we show how questions can be automatically generated on the basis of a given configuration knowledge base. In Section 5 we present the results of a qualitative study related to the applicability of STUDYBATTLES. In Section 6 we discuss future research issues and conclude the paper.

## 2 STUDYBATTLES

The STUDYBATTLES[4] start screen is shown in Figure 1 – it includes a list of subscriptions to learning applications (*LearnApps*) and further information regarding the ranking of the user in specific learning applications. Mobile clients for STUDYBATTLES are available in *Android*, *iOS*, and *HTML-5* – Figure 1 depicts an example screenshot of an *iOS version*. The system can be deployed in a company's intranet and is also available as global server solution.[5] Users can join communities and subscribe to learning applications in which they can add learning content, practice exercises, and compete against other learning application users in a (quiz-based) duel. Contents within learning applications are organized in terms of categories, for example, the learning application "*Master Of Configuration*" includes the categories *Sales Knowledge*, *Conflicts*, *Diagnosis*, *Incompatibilities*, *Knowledge Acquisition*, and *Knowledge Representation*.

*Deployments of* STUDYBATTLES. One version of the system has already been deployed and is applied by a large municipality and two universities in Austria. At the two mentioned universities, STUDYBATTLES is applied in three *Software Engineering* courses (*Object-oriented Analysis and Design*, *Software Paradigms*, and *Requirements Engineering*) and in two *Artificial Intelligence* related courses (*Configuration Systems* and *Recommender Systems*). The goal of the STUDYBATTLES instance deployed at one Austrian municipality is to increase employee's knowledge in security-related topics and also 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 goal in this context is to support sales representatives in learning processes related to *product knowledge and sales practices*. Experts from these domains participated in a qualitative study where they gave feedback on system applicability (see Section 5).



Applied Software Engineering Practi... (2.)
ASE (87%)

BugFixing (1.)
BF (0%)

Group Decision Making (1.)
GROUP (0%)

Internet and new Media (1.)
INM (30%)

Master Of Configuration (7.)
MOC (17%)

Object-Oriented Analysis & Design (2.)
Objects (6%)

Recommender Systems (2.)
RecSys (0%)

**Figure 1.** STUDYBATTLES start screen (iOS version) consisting of learning applications (*LearnApps*) that can be subscribed by the user. Percentages report the share of already successfully answered questions. "*Master Of Configuration*" is the learning application that includes configuration-related knowledge.

*Learning and training*. After a STUDYBATTLES learning application has been subscribed, users of this application can select categories and questions they want to answer. After having selected an answer to a question, immediate feedback is provided on the correctness of the answer. If an answer is wrong, related explanations can be provided to the user. Explanations can only be shown if these have been included by the expert who entered a question and related answers, i.e., in the current version of STUDYBATTLES explanations are not determined automatically.

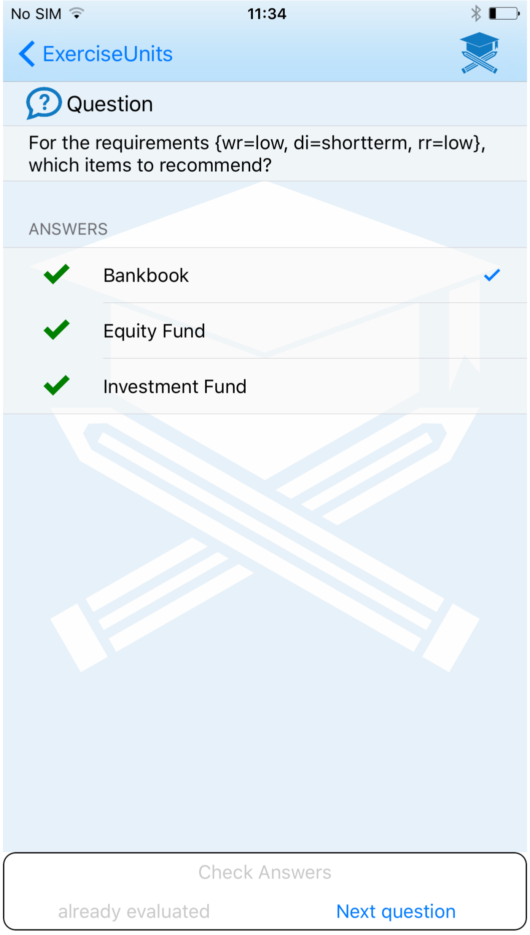
*Content creation and question types*. STUDYBATTLES follows the concept of *crowd sourcing* where users can enter questions/answers

[4] STUDYBATTLES has been developed within the scope of the PEOPLEVIEWS research project which is funded by the Austrian Research Promotion Agency (843492).

[5] www.studybattles.com.

and expert users can evaluate the quality of the questions. The status of a *domain expert* is reached if a certain treshold of correctly answered questions is passed. Users are allowed to add additional content in terms of documents, pictures, and movies which serve as a basis for answering questions. When a user interacts with a learning application, questions are recommended [10] depending on their relevance for the user. Questions related to a category where a user has a low learning performance have a higher probability of being recommended to the user.

STUDYBATTLES supports the definition of different types of questions – examples thereof will be discussed in the following. Figure 2 depicts an example of a *multiple-choice question* that is related to the category *Sales Knowledge*. This question reflects relationships between customer requirements and financial services (equity fund, investment fund, and bankbook). The used abbreviations reflect the set of customer requirements $\{wr$ = *willingness to take risks*, $di$ = *duration of investment*, and $rr$ = *expected return rate*$\}$.

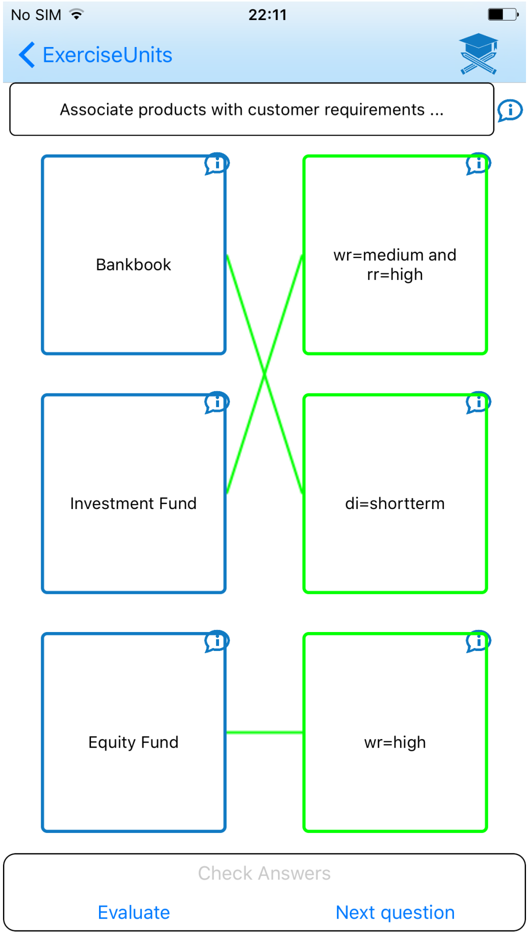


For the requirements {wr=low, di=shortterm, rr=low}, which items to recommend?

**Figure 2.** STUDYBATTLES: representation of multiple-choice questions – the check mark on the right hand side represents the answer of the user, check marks on the left hand side represent the correctness feedback.

An example of an *association task* is depicted in Figure 3 – the corresponding HTML-based definition interface is depicted in Figure 9. In association tasks, terms on the right-hand side have to be combined (associated) with the terms on the left-hand side. In the example of Figure 3, association tasks are exploited for asking questions that are related to the compatibility of customer requirements and products. Association tasks can also be applied to ask questions, for example, about the incompatibility of specific customer requirements. In the example of Figure 4 users are requested to combine individual customer requirements of the left and right hand side in such a way that the connected requirements become *inconsistent*. In Figure 5, a question is posed to educate users with regard to logical entailment (which situations lead to an empty set of solutions). In this example, only a high willingness to take risks is logically entailed in the item *Equity Fund*.

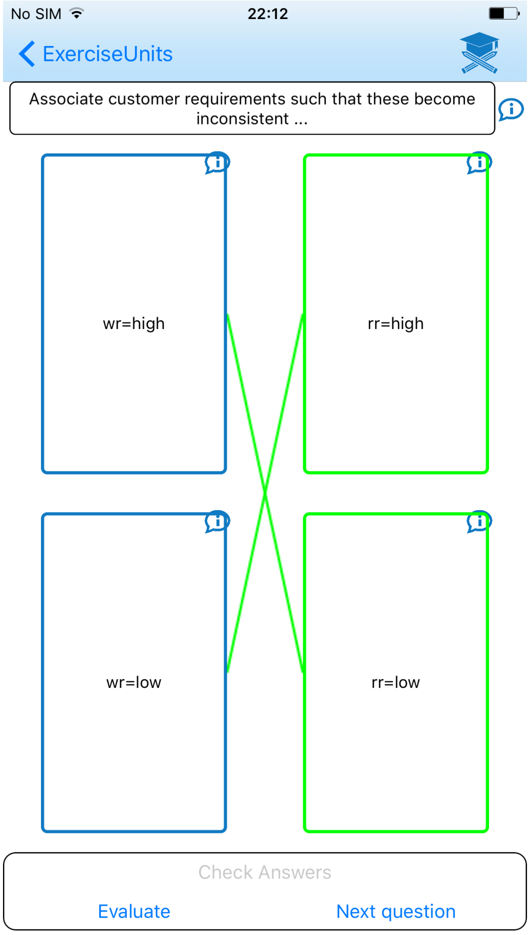
Figure 6 includes a question that is related to the correct usage of implications [6]. If a certain constraint is specified on a textual level (e.g., *a low willingness to take risks can only be combined with a bankbook*), the corresponding logical representation should be clear for all knowledge engineers. In order to avoid faulty translations, exercises such as the example depicted in Figure 6 can help to establish a standard of formalizing such properties. Finally, Figure 7 depicts an example question related to the identification of redundant constraints in configuration knowledge bases [4]. A constraint is considered as redundant if its deletion from the knowledge base is semantic-preserving, i.e., the solution space remains the same. On the logical level a constraint $c_a \in C$ is considered redundant if $C - c_a \models c_a$.



Associate products with customer requirements ...

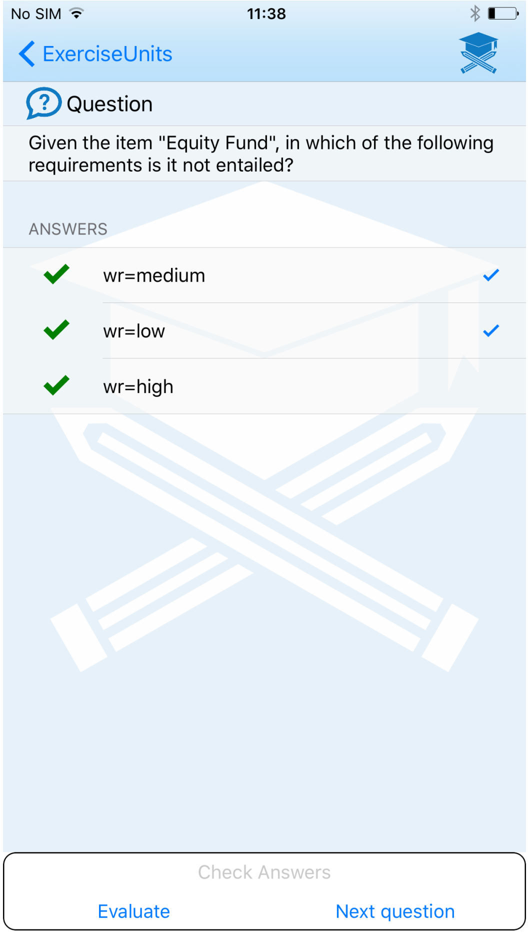
**Figure 3.** STUDYBATTLES: representation of associations tasks (concepts on the left have to be connected with the correct counterparts on the right).

*Gamification.* Users who trigger duels are then randomly assigned



Associate customer requirements such that these become inconsistent ...

**Figure 4.** STUDYBATTLES: Association task related to the association of incompatible requirements.



Given the item "Equity Fund", in which of the following requirements is it not entailed?

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

to opponents – duels can be performed asynchronously. User ranking is visible on a local level (users are only able to see opponents directly ranked before or after them in a certain learning application). If a user wins a duel, he/she receives corresponding STUDYBATTLES POINTS which is a major motivation for users to engage in games. The higher the complexity of an answered question, the higher the amount of received STUDYBATTLES POINTS. The complexity of a question can be evaluated directly after having answered the question (see the *Evaluate* link, for example, in Figure 5).

*Analysis of learning performance*. STUDYBATTLES supports different types of statistics 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). In each learning application, each user has access to a ranking where the "direct-neighbor" opponents including their STUDYBATTLES POINTS are shown. Administrators of a STUDYBATTLES community have access to statistics that indicate the overall learning performance per learning application and also per topic inside a learning application. This way, strengths and weaknesses of a STUDYBATTLES learning community can be identified and corresponding counter-measures, for example, in terms of improving specific learning contents, can be triggered.

## 3 Example Configuration Knowledge Base

As a working example we introduce a simplified financial services configuration task. Before introducing the example, we provide a basic definition of a configuration task and a corresponding solution (configuration).

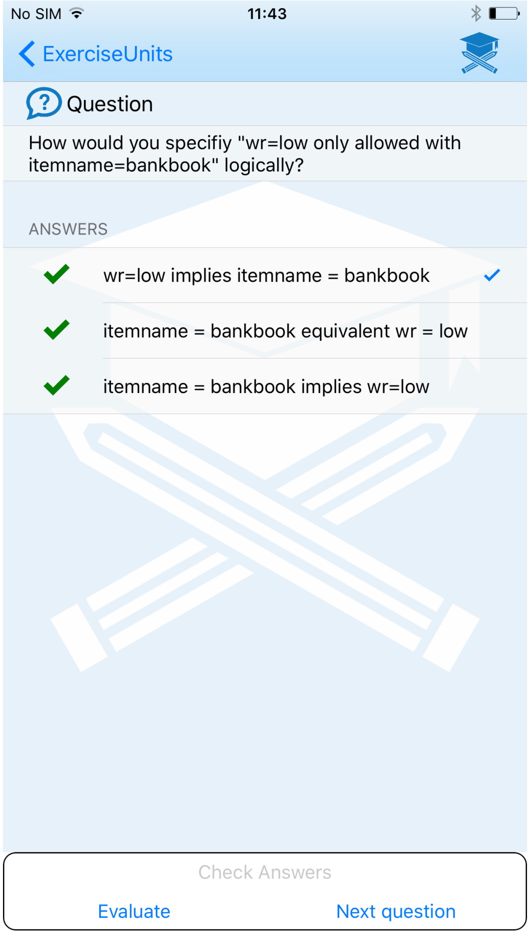
*Definition 1 (Configuration Task).* A configuration task can be defined as a Constraint Satisfaction Problem (CSP) ($V$, $D$, $C$, $REQ$) where $V$ are variables, $D$ are domain definitions for the variables, $C$ is a set of constraints[6], and $REQ$ is a set of customer requirements.

*Definition 2 (Configuration).* A configuration (solution) for given configuration task ($V$, $D$, $C$, $REQ$) is a complete set $conf$ of variable assignments $v_i = a$ to the variables $v_i \in V$ ($v_i = a \rightarrow a \in$ domain($v_i$)) with *consistent*($conf \cup C \cup REQ$). The set of solutions for a given configuration task is denoted as $CONFS$.

Configurations (one configuration task can have more than one solution) can be ranked according their utility for the user (customer). In this context, configurations with the highest utility for the user can be regarded as recommendations – for details we refer to [4].

The following is a simple *financial services configuration knowledge base* formulated as configuration task (see Definition 1). The variables in $V$ are the following: *willingness to take risks* ($wr$), *du-*

[6] Note that $C = \{c_1, c_2, ..., c_n\}$ can also be represented as $C = \{c_1 \wedge c_2 \wedge ... \wedge c_n\}$ since constraints are assumed to be connected conjunctively.



How would you specifiy "wr=low only allowed with itemname=bankbook" logically?

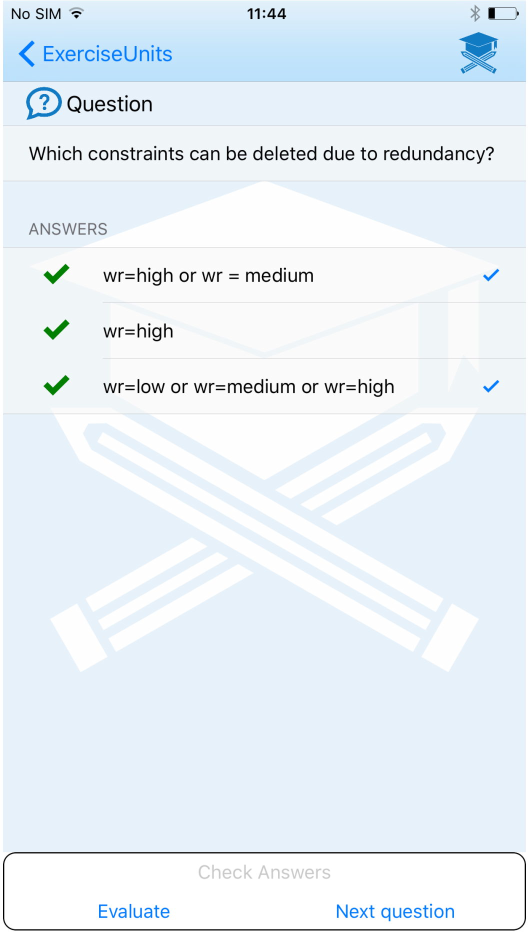
ANSWERS

✓ wr=low implies itemname = bankbook ✓

✓ itemname = bankbook equivalent wr = low

✓ itemname = bankbook implies wr=low

**Figure 6.** STUDYBATTLES: question related to the usage of implications.



Which constraints can be deleted due to redundancy?

ANSWERS

✓ wr=high or wr = medium ✓

✓ wr=high

✓ wr=low or wr=medium or wr=high ✓

**Figure 7.** STUDYBATTLES: question related to the identification of redundant constraints in configuration knowledge bases.

*ration of investment* ($di$), *expected return rate* ($rr$), and $itemname$ is a variable which represents the name of a financial service.

- $V = \{wr, di, rr, itemname\}$
- $D$ = {domain($wr$) = {low, medium, high}, domain($di$) = {shortterm, mediumterm, longterm}, domain($rr$) = {low, medium, high}, domain($itemname$) = {equityfund, investmentfund, bankbook}}
- $C = \{c_1 : \neg wr = low \vee itemname = bankbook, c_2 : \neg wr = medium \vee itemname \neq equityfund, c_3 : \neg di = shortterm \vee itemname = bankbook, c_4 : \neg di = mediumterm \vee itemname \neq equityfund, c_5 : \neg(rr = high \vee rr = medium) \vee itemname \neq bankbook, c_6 : \neg(wr = low \wedge rr = high), c_7 : \neg(di = shortterm \wedge rr = high), c_8 : \neg(wr = high \wedge rr = low)\}$
- $REQ = \{r_1 : wr = low, r_2 : di = shortterm, r_3 : rr = low\}$

A configuration for our example configuration task is the set of variable assignments $conf = \{wr = low, di = shortterm, rr = low, itemname = bankbook\}$ since the customer requirements included in $REQ$ are consistent with the constraints in $C$. If we change the specification of $REQ$ this can lead to situations were requirements become inconsistent with the constraints in $C$, i.e., no solution can be found. Such a situation is triggered in the case that $REQ = \{r_1 : wr = low, r_2 : di = shortterm, r_3 : rr = high\}$, i.e., $REQ \cup C$ is inconsistent. In such situations, model-based diagnosis [13] can be exploited to identify minimal sets of requirements that have to be adapted or deleted such that a solution (configuration) can be found. The identification of such adaptations can be formulated as diagnosis task (see Definitions 3–4).

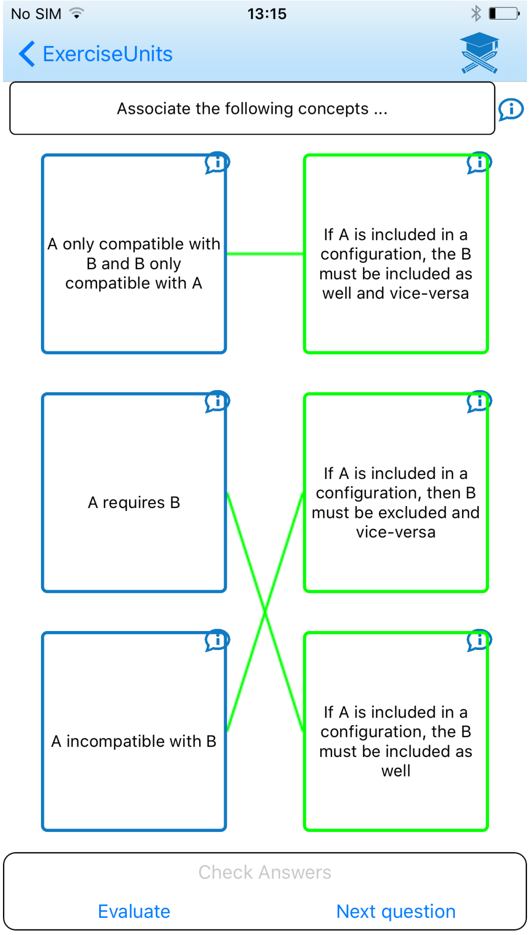
*Definition 3 (Diagnosis Task).* A diagnosis task is defined as a tuple ($C$, $REQ$) where $C$ is a set of constraints, $REQ$ is a set of customer requirements, and $REQ \cup C$ is inconsistent.

*Definition 4 (Diagnosis).* A set $\Delta \subseteq REQ$ for a given diagnosis task ($C$, $REQ$) is a diagnosis if $REQ - \Delta \cup C$ is consistent, i.e., $\Delta$ is a set of requirements to be deleted from $REQ$ such that consistent($REQ - \Delta \cup C$). $\Delta$ is *minimal* if $\neg\exists\Delta'$: $\Delta' \subset \Delta$.

Diagnoses are often denoted as hitting sets [13]. The original algorithm for determining minimal hitting sets is introduced in [13].

Finally, conflicts (also denoted as conflict sets) represent sets of requirements (in $REQ$) that are able to induce an inconsistency with $C$ (see the following definition). Conflict sets can be exploited for the determination of diagnoses but also for the determination of redundant constraints in knowledge bases (see, e.g., [4]). There is a duality between conflicts and diagnoses: a conflict set is a hitting set for a set of minimal diagnoses and – vice versa – a diagnosis is a hitting set for a set of minimal conflicts [4].

*Definition 5 (Conflict Set).* A set $CS \subseteq REQ$ is a conflict set if $CS \cup C$ is inconsistent. $CS$ is *minimal* if $\neg\exists CS'$: $CS' \subset CS$.



Associate the following concepts ...

A only compatible with B and B only compatible with A

If A is included in a configuration, the B must be included as well and vice-versa

A requires B

If A is included in a configuration, then B must be excluded and vice-versa

A incompatible with B

If A is included in a configuration, the B must be included as well

**Figure 8.** STUDYBATTLES: association task related to the modeling concepts of a configuration language.

*Types of sales knowledge*. Sales knowledge is included in different forms in configuration knowledge bases. A configurator can determine items that can be recommended (*product knowledge*) on the basis of a given set of customer requirements ($REQ$). In situations where no solution can be identified for a given set of customer requirements, diagnosis algorithms [13] can determine the needed minimal changes to help the user out of the *no solution could be found dilemma* (*analysis knowledge*). Finally, given a set of requirements (REQ), a conflict detection algorithm [11] can determine a minimal set of customer requirements which induce an inconsistency with $C$ (*inconsistency knowledge*). In the following we discuss how configuration task definitions can be exploited for the automated generation of related questions for STUDYBATTLES.

*Types of engineering knowledge*. Engineering knowledge entails relevant practices when building and maintaining configuration knowledge bases. Knowledge engineers need to understand how to *formalize* different product related constraints (e.g., how should requirements relationships be specified on a logical level – see Figure 6). It is also important to understand *well-formedness criteria* relevant for knowledge base development and maintenance (e.g., what are redundant constraints – see Figure 7). Furthermore, knowledge engineers need to know the building blocks of the language used for configuration knowledge representation. A simple related STUDYBATTLES question is depicted in Figure 8.

## 4 Generating Questions from Configuration Knowledge Bases

Some of the STUDYBATTLES questions can be automatically generated from a configuration task definition. In the following we show how questions (and related answers) can be generated for some of the configuration knowledge types discussed in Section 3. Generated questions can be included in STUDYBATTLES and then used for training purposes. These questions can be exploited by employees to improve their configuration-related knowledge.

*Product knowledge*. The task of a user is to identify configurations that are consistent with a given set of customer requirements ($REQ = \{r_1, r_2, ..., r_m\}$). Related correct and faulty answers can be generated by a constraint solver on the basis of a configuration task ($V$, $D$, $C$, $REQ$). A constraint solver calculates configurations that satisfy $REQ \cup C$. Furthermore, *non-solutions* satisfy $REQ \cup \overline{C}$.[7] $REQ = \{r_1, r_2, ..., r_m\}$ then is the basis of a question, *solutions* represent correct answer(s), and *non-solutions* represent faulty answer(s). Since the potential number of solutions and non-solutions can be high, a random number thereof is selected for inclusion in STUDYBATTLES.

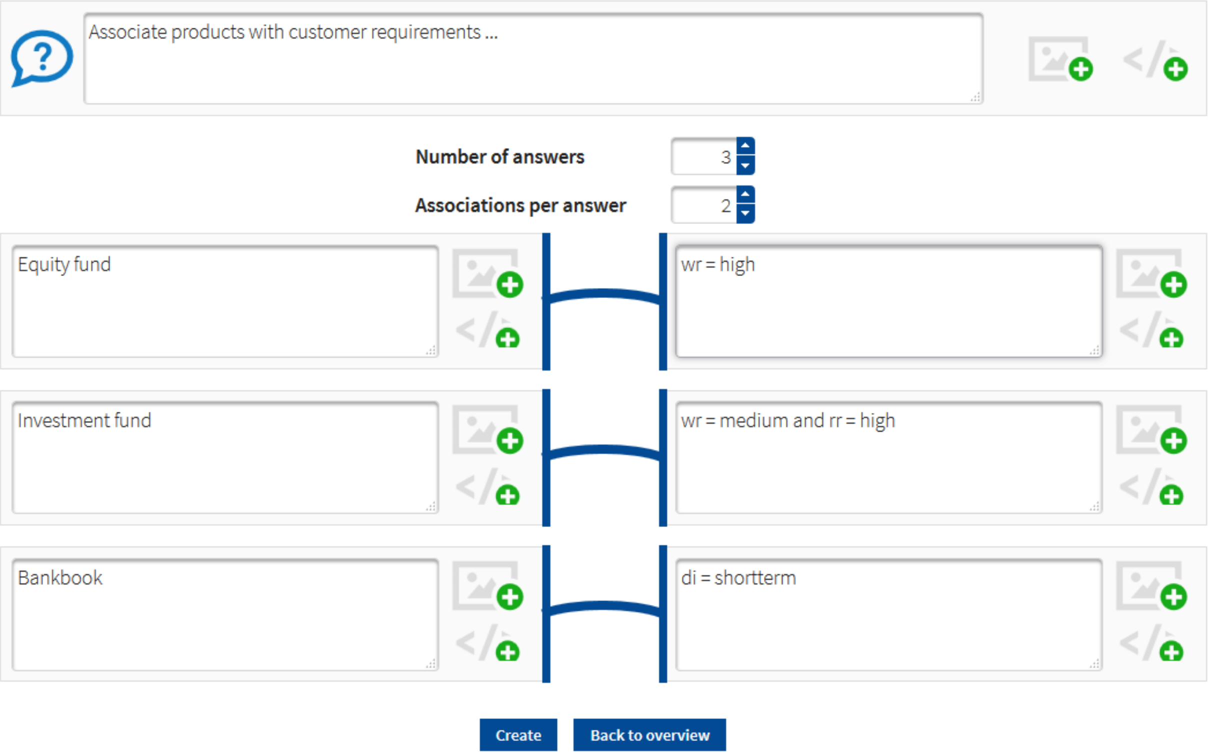
*Product knowledge (example)*. Given our example configuration task definition, customer requirements could be $REQ = \{r_1 : wr = low, r_2 : di = shortterm, r_3 : rr = low\}$, a solution (correct answer) is $\{name = bankbook\}$, and non-solutions are $\{name = equityfund, name = investmentfund\}$ (see Figure 2). A related question posed in STUDYBATTLES is: *For the requirements ..., which items to recommend?*

*Analysis knowledge*. Assuming that $REQ \cup C$ is inconsistent (and $C$ is consistent), the task is of a user is to figure out which minimal set of $r_i \in REQ$ has to be deleted such that consistency can be restored. More formally, analysis knowledge related questions can be generated using a diagnosis task ($C, REQ$). The diagnosis task definition ($C, REQ$) can be used for question representation, related answers are represented by the diagnoses $\Delta_i$. Non-diagnoses can be easily determined on the basis of a calculated set of diagnoses: if, for example, $\Delta_i = \{r_a, r_b, r_c\}$ is a minimal diagnosis, then $\Delta_{in} = \{r_a, r_b\}$ is a corresponding non-diagnosis since a proper subset of minimal diagnosis is not a diagnosis. Since the potential number of diagnoses and non-diagnoses can be high, the answer set is composed of a random number of selected diagnoses and non-diagnoses.

*Analysis knowledge (example)*. Given our configuration task definition with the customer requirements $REQ = \{r_1 : wr = low, r_2 : di = shortterm, r_3 : rr = high\}$. Alternative minimal sets of customer requirements (diagnoses $\Delta_i$) that have to be deleted from $REQ$ such that a solution can be identified, are: $\{\Delta_1 = \{r_1, r_2\}, \Delta_2 = \{r_3\}\}$, i.e., deleting the requirements $r_1$ and $r_2$ restores consistency between $REQ$ and $C$. An example of a non-diagnosis related to diagnosis $\Delta_1$ is $\{r_1\}$. A related question posed in STUDYBATTLES is: *Given the configuration task definition ... which one is a minimal set of requirements that have to be deleted from REQ such that consistency can be restored?*

*Inconsistency knowledge*. Assuming that $REQ \cup C$ is inconsistent (and $C$ is consistent), the task is of a user is to figure out which minimal set of $r_i \in REQ$ is inconsistent with $C$. More formally, inconsistency knowledge related questions can be generated on the basis of a conflict detection task ($C, REQ$). The conflict detection task ($C, REQ$) can be used for question representation, related correct answers are represented by the conflict sets $CS_i$. Non-conflicts (faulty answers) can be easily determined on the basis of a calculated

[7] If $C = \{c_1 \wedge c_2 \wedge ... \wedge c_n\}$ then $\overline{C} = \{\neg c_1 \vee \neg c_2 \vee ... \vee \neg c_n\}$.



Associate products with customer requirements ...

Number of answers 3

Associations per answer 2

| | |
|---|---|
| Equity fund | wr = high |
| Investment fund | wr = medium and rr = high |
| Bankbook | di = shortterm |

**Figure 9.** StudyBattles: definition of an association task (specification of correct pairs) in the HTML-5 version.

set of conflicts: if, for example, $CS_i = \{r_a, r_b\}$ is a minimal conflict, then $CS_{in} = \{r_a\}$ is a non-conflict since subsets of minimal conflicts do not have the conflict property. Since the potential number of conflicts and non-conflicts can be high, the answer set is composed of a random number of selected conflicts and non-conflicts.

*Inconsistency knowledge (example).* Given our configuration task definition with the customer requirements $REQ = \{r_1 : wr = low$, $r_2 : di = shortterm$, $r_3 : rr = high\}$. Alternative minimal sets of customer requirements subset of $REQ$ that are inconsistent with $C$ are: $\{CS_1 = \{r_1, r_3\}\}$ and $\{CS_2 = \{r_2, r_3\}\}$. An example of a non-conflict related to conflict set $CS_1$ is $\{r_1\}$. A related question posed in StudyBattles is: *Given the configuration task definition ... which one is a minimal set of requirements that have to be deleted from $REQ$ such that consistency can be restored?*

For questions related to the *formalization* of informal domain descriptions and constraints (see Figure 6) we do not offer an automated question generation mechanism. The same holds for *modeling concepts* for the development and maintenance of configuration knowledge bases. Questions related to *well-formedness criteria* for the development of configuration knowledge bases can be automatically generated. For example, a configuration knowledge base containing redundant constraints (the question part) can be presented to the user. The correct answers (redundant constraints) can be identified by corresponding redundancy detection mechanisms – for details we refer to [4]. Other examples of well-formedness rules are discussed in [3].

## 5 StudyBattles Evaluation

StudyBattles has been evaluated within the scope of a qualitative study (N=15 participants). Participants from different domains (financial services, public administration, telecommunications, and universities) provided feedback on the applicability and usefulness of StudyBattles. Major mentioned potential improvements that come along with StudyBattles are, improved knowledge retention in organizations, 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, increased learning efficiency, and enhanced possibilities of community knowledge analysis which provide a basis for a fine-grained adaptation of learning material.

The *application of an e-learning environment in configuration scenarios* was motivated by discussions with different companies applying configuration technologies. Especially in distributed scenarios where large-scale configuration knowledge bases have to be developed and maintained, additional learning mechanisms have to be provided to establish a standard knowledge level that helps to reduce erroneous maintenance practices as well as suboptimal sales practices. A major issue in this context is that existing commercial configuration environments still do not provide intuitive knowledge representation mechanisms and there is a need to further educate domain experts and knowledge engineers.

## 6 Conclusions and Future Work

In this paper we introduced the idea of applying an e-learning environment (StudyBattles) as a complementary approach to transfer configuration-related knowledge to employees (e.g., sales representatives and knowledge engineers). We provided an overview of different system functionalities such as community-based content development, gamification, and automated question generation. In the context of automated question generation we focused on the two aspects of generating sales and engineering related knowledge. More fine-grained question generation techniques that provide mechanisms to more systematically pre-select answers to be included are within the scope of future work. In this context we will also analyze potential synergies with existing approaches to test case generation in software engineering [2]. Especially in the context of educating sales representatives, automated question generation becomes a key functionality, since this 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).

## REFERENCES

[1] M. Aldanondo and E. Vareilles, 'Configuration for mass customization: how to extend product configuration towards requirements and process configuration', *Journal of Intelligent Manufacturing*, **19**(5), 521–535, (2008).

[2] S. Anand, E. Burke, T. Chen, J. Clark, M. Cohen, W. Grieskamp, M. Harman, M. Harrold, and P. Mcminn, 'An orchestrated survey of methodologies for automated software test case generation', *Journal of Systems and Software*, **86**(8), 1978–2001, (1987).

[3] D. Benavides, S. Segura, and A. Ruiz-Cortez, 'Automated analysis of feature models 20 years later: A literature review', *Information Systems*, **35**(6), 615–636, (2010).

[4] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Elsevier/Morgan Kaufmann Publishers, 1st edn., 2014.

[5] A. Felfernig, M. Jeran, T. Ruprechter, A. Ziller, S. Reiterer, and M. Stettinger, 'Learning games for configuration and diagnosis tasks', in *16th International Workshop on Configuration*, pp. 111–114, Vienna, Austria, (2015).

[6] A. Felfernig, S. Reiterer, M. Stettinger, and J. Tiihonen, 'Towards understanding cognitive aspects of configuration knowledge formalization', in *Vamos 2015*, pp. 117–124, Hildesheim, Germany, (2015).

[7] BJ Fogg, *Persuasive Technology*, Morgan Kaufmann Publishers, 1st edn., 2003.

[8] 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).

[9] L. Hvam, N. Mortensen, and J. Riis, *Product Customization*, Springer, 1st edn., 2010.

[10] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, Cambridge University Press, New York, NY, USA, 1st edn., 2010.

[11] U. Junker, 'Quickxplain: Preferred explanations and relaxations for overconstrained problems', in *19th National Conference on Artificial Intelligence (AAAI04)*, pp. 167–172, (2004).

[12] S. Rakangor and Y. Ghodasara, 'Literature review of automatic question generation systems', *International Journal of Scientific and Research Publications*, **5**, (2015).

[13] R. Reiter, 'A theory of diagnosis from first principles', *AI Journal*, **23**(1), 57–95, (1987).

[14] A. Felfernig S, Reiterer, M. Stettinger, F. Reinfrank, M. Jeran, and G. Ninaus, 'Recommender systems for configuration knowledge engineering', in *Workshop on Configuration*, pp. 51–54, Vienna, Austria, (2013).

[15] M. Stumptner, 'An Overview of Knowledge-based Configuration', *AI Communications*, **10**(2), 111–126, (1997).

[16] D. Zhang and J. Nunamaker, 'Powering E-Learning In the New Millenium: An Overview of E-Learning and Technology', *Information Systems Frontiers*, **5**(2), 207–218, (2003).