Multi Category Content Selection in Spaced Repetition Based Mobile Learning Games

Florian Schimanke  
Dept. of Computer Science  
HSW University of Applied Sciences  
Hameln, Germany  
schimanke@hsw-hameln.de

Oliver Vornberger  
Dept. of Mathematics & Computer Science  
University of Osnabrueck  
Osnabrueck, Germany  
oliver@uos.de

Robert Mertens  
Dept. of Computer Science  
HSW University of Applied Sciences  
Hameln, Germany  
mertens@hsw-hameln.de

Stephanie Vollmer  
Dept. of Business Science  
HSW University of Applied Sciences  
Hameln, Germany  
vollmer@hsw-hameln.de

Abstract—Learning requires repetition. Spaced repetition algorithms are aimed at reducing the number of times a learning item has to be accessed by the learner by scheduling item presentation based on psychological models. These models take into account learner performance on previous interactions with the learning item and the rate at which humans forget what they have learned. In recent years, spaced repetition learning software has become popular for simple learning tasks like flash cards used for learning vocabulary. This paper presents a prototype application that extends the spaced repetition learning approach to more complex content like the kind usually found in learning games. One major difference between this content and flash cards is that learning games usually contain a number of different tasks that convey the same underlying concept categories. To complicate matters, one task might even be classified as belonging to a number of independent or orthogonal categories. This paper explores how these categories can be modeled on the basis of a mobile game designed for training in the field of relational databases. We have chosen a mobile approach to leverage it’s anytime/anyplace availability which allows a more precise scheduling by the spaced repetition algorithm.

Keywords: game based learning, mobile learning, spaced repetition, SuperMemo, learning task categorization

I. INTRODUCTION

In a time where smartphones and tablets are omnipresent, learning with those devices becomes ubiquitous. Having these devices always at hand enables learners to learn with them almost anywhere and anytime and also fosters the spaced repetition learning approach. This technique is based on the spacing effect [12] and aims to determine the best time between learning intervals to achieve the best possible result and hence the best relation between time spent learning and learning success. Studies have shown that repeating learning content at certain intervals has a huge effect on long-term memorization. According to Kornell, this is also far more effective than learning with other strategies like massing or cramming [6]. By using computers, algorithms can help learners in determining the best assumed intervals between each learning period. Based on the results the algorithm is also able to schedule repetitions for good remembered content farther in the future than repetitions for less good remembered content. With the spreading of mobile devices, these algorithms become ubiquitously available. Learning apps and mobile games can make use of them, which enables learners to also use spaced repetition learning while on the go. Most mobile devices are also capable of presenting alerts at a given time and thus reminding the learner when it is time for learning.

While some topics like vocabulary should be learned on an atomic base, i.e. every word for itself, other learning areas can be divided into categories which subsume the content into certain areas they belong to. Projecting the spaced repetition approach onto this categorization means a certain adjustment of the currently used traditional algorithms to determine the best learning intervals. Using this in learning games also requires an additional, round-based algorithm, which helps to avoid a corruption of the original, time-based spaced repetition algorithm when playing the same content several times in a row, while ensuring that the game does not get boring because of back to back repetitions of the same content.

While most learners perceive learning as something annoying, games offer an engaging and motivating way to learn. In fact, the NMC Horizon Report ranks the time-to-adoption horizon for game-based learning at two-to-three years [4]. This paper introduces a game for learning the
As one can see, games have the potential to trigger a cycle that can enhance the learning outcomes. This cycle includes a continuous loop of user judgments, user behavior, and system feedback. After having injected the instructional content and the game characteristics, the learner judges what he gets presented. The result of that judgment should be things like interest or enjoyment, which keeps him playing several rounds of the game and re-playing it in the future. This can lead to user behaviors like improved times to solve a task or greater persistence which may then influence the system feedback. If the game is designed properly, this cycle may result in self-motivated learning. After having played the game several times, the learner is able to apply the learned objectives, which corresponds to the concept of “doing, reflecting, understanding, and applying” as described by Kolb [5].

This concept may very well fit together with spaced repetition learning. The latter is already commonly used in fields like vocabulary training, which often relies on flashcards. The idea behind spaced repetition is to find an optimum interval between the repetitions of one learning item. According to Bahrick and Phelps these optimum intervals are the longest possible intervals that do not lead to forgetting [1]. From that Wozniak [10] concludes two principles to calculate the optimum intervals:

1. “Intervals should be as long as possible to obtain the best learning results, the intervals between repetitions of the same card should increase the better the learner remembers the correct answer [11]. To calculate the intervals between learning sessions there are already several algorithms and applications available. One of the most well known representations of them is SuperMemo and its SuperMemo2 (SM2) algorithm1 which also lays the groundwork for other algorithms like the ones used in Mnemosyne2 or Anki3. However, these algorithms are only used in flashcard-based learning. While there are already mobile implementations of these algorithms, for example an app called Repetitions4, it is not currently used in game-based learning.

There are, however, already several mobile learning games available that use another approach, which often focuses more on imparting knowledge rather than immersing it. While imparting knowledge focuses just on providing learning sessions there are already several algorithms and applications available. One of the most well known representations of them is SuperMemo and its SuperMemo2 (SM2) algorithm1 which also lays the groundwork for other algorithms like the ones used in Mnemosyne2 or Anki3. However, these algorithms are only used in flashcard-based learning. While there are already mobile implementations of these algorithms, for example an app called Repetitions4, it is not currently used in game-based learning.

Furthermore it was shown that in order to achieve the best learning results, the intervals between repetitions of the same card should increase the better the learner remembers the correct answer [11]. To calculate the intervals between learning sessions there are already several algorithms and applications available. One of the most well known representations of them is SuperMemo and its SuperMemo2 (SM2) algorithm1 which also lays the groundwork for other algorithms like the ones used in Mnemosyne2 or Anki3. However, these algorithms are only used in flashcard-based learning. While there are already mobile implementations of these algorithms, for example an app called Repetitions4, it is not currently used in game-based learning.

There are, however, already several mobile learning games available that use another approach, which often focuses more on imparting knowledge rather than immersing it. While imparting knowledge focuses just on providing learning sessions there are already several algorithms and applications available. One of the most well known representations of them is SuperMemo and its SuperMemo2 (SM2) algorithm1 which also lays the groundwork for other algorithms like the ones used in Mnemosyne2 or Anki3. However, these algorithms are only used in flashcard-based learning. While there are already mobile implementations of these algorithms, for example an app called Repetitions4, it is not currently used in game-based learning.

Figure 1: Input-Process-Outcome Game Model acc. to Garris [3]

II. RELATED WORK

Game-based learning is often seen as a way to enhance the learning experience and to foster the understanding of complex subjects as was shown in empirical research by Cordova & Lepper [2] or Ricci, Salas, & Cannon-Bowers [8]. One key aspect for using game-based learning is that it provides a way to keep the motivation of the learner at a high level. According to Ricci et al. games enhance learner motivation, which leads to an improved attention and retention of the respective topics. The actual content is wrapped into a game scenario and learned in a more abstract and engaging way. After having understood the topic, the learner should then be able to apply the learned knowledge in real-world scenarios. Based on this, Garris [3] has developed the “Input-Process-Outcome Game Model” to be seen in Figure 1.

The remainder of the paper is organized as follows. In the related work section we present the status quo of related fields of research with a focus on game based learning and the way learning games might benefit from content selection algorithms before we describe the approach introduced in this paper. After that we give an overview over categorization of learning contents and how this categorization might occur. This categorization is then implemented in our prototype learning game where it makes use of the content selection algorithms. After showing our prototype app we discuss the pros and cons and give a forecast for future work.

1 http://www.supermemo.com
2 http://www.mnemosyne-proj.org
3 http://ichi2.net/anki
4 https://itunes.apple.com/de/app/repetitions-for-iphone-ipod/id332352818?mt=8
learner’s motivation by avoiding to select the same content several times in a row. Since the model is designed as a cycle the injection of the re-scheduling can take place continuously after a certain amount of rounds.

There are several different approaches which are seen as decisive game characteristics. One of those approaches is “trial and error” or “learning by mistake” as described by Prensky [7]. When using this approach it is important that the learner gets a feedback about his mistake so that he can learn from it and does not make it again. Combining this idea with spaced repetitions, the learner should make less mistakes the more often he plays the game and gets the appropriate feedback.

III. SYSTEM DESCRIPTION

In an earlier paper we have already introduced a way to transfer the SM2 algorithm for scheduling spaced repetitions to game-based learning [9]. The app “Where is my Box?” was intended to show how spaced repetition language learning can be done with a mobile learning game. While evaluating this prototype it has come to light that using only the SM2 algorithm in this scenario is not appropriate since it is a strictly time-based solution. This fits perfectly in scenarios where there is a lot of content, for example when using flashcards with vocabulary. On the other hand, when using game-based learning, content creation usually takes more effort as the game should consist of a number of challenging and interesting tasks. Hence, there is usually a limited amount of content which often leads the user to play several rounds of the game in a row. This might corrupt the spaced repetition concept since the calculated time by the algorithm is not elapsed before the learner starts the next repetition. It might also affect the playing behavior since reminders for a repetition can get pushed back too far into the future which might keep the learner from using the game in the best possible intervals as calculated by the algorithm. Therefore we have developed another algorithm (the FS algorithm) which is round-based and which takes over the content selection when the user decides to play one or more rounds after the initial, scheduled one. Herewith we are also able to avoid too frequent repetitions of the same content and also focus on topics that the learner does not already know very well.

The FS algorithm which was introduced in the paper mentioned above does exactly that. After the SM2 algorithm has selected the appropriate content after starting the game and rescheduled the next repetition, the FS algorithm is in charge to mimic its concept of ranking learning content by the level the learners already remembers it in a round-based manner for the remainder of that game session. From the moment the FS algorithm takes over, there are no more values manipulated or reschedulings done which would affect the integrity of the SM2 algorithm.

The new prototype app implements a game for learning the entity-relationship model of relational databases. It makes use of both algorithms to provide a scheduling of repetitions according to the spaced repetition concept.

IV. CATEGORIZATION OF LEARNING CONTENT

As mentioned, the groundwork for this game as well as the game introduced in this paper is built on the idea of spaced repetition. One problem with using this idea in learning fields other than vocabulary is that learning contents may be very similar. While in vocabulary learning every word is a single content for itself, complex games may use several contents from the same topic. This raises two considerations. First, the game should not get boring after a short time due to repeating the same content over and over again. And second, the game should foster learning concepts rather than facts. Therefore we decided to aggregate similar content items in categories. This leads to the effect that the scheduling of future repetitions is based on categories instead of atomic content. By doing so we are able to avoid too frequent repetitions of similar contents and instead focus on the actual learning topic.

However, categorization of learning content does not make sense in every context except for avoiding too frequent repetitions of the same content. But in other cases categorization can be very useful. An example might be to learn the capitals of several countries all over the world. While a European might be able to know most of the capitals and their position from European countries, it might be much harder to learn the capitals of African or Asian countries. This can lead to a categorization of the countries and their capitals for example by continents (i.e. Europe, Africa, Asia, …) or even more broken down by region (i.e. North-Africa, West-Africa, …).

In our example we have built a mobile game for learning entity-relationship modeling in relational databases. There are several different learning topics which can be categorized in this field of learning. We have chosen three topics to deal with in our game. First, database diagrams use different cardinality constraints (i.e. one-to-one, one-to-many, many-to-many, etc.) to reflect the relationships between the different entities. They may be extended by conditional relations (i.e. 1:NC). These constraints can be used on each side of the relationship. They can be the same on each side or they can differ and entities can have several relationships with other entities and with different cardinality constraints. In a more complex entity-relationship model there can be several different cardinalities all over the place. Therefore, we have decided to categorize them by the four main constraints according to the crow’s foot notation which was developed by James Martin, Bachmann und Odell, extended with conditional relations.

Another topic with entity-relationship modeling is dealing with entities. This is our second category of learning content. An entity in data modeling is a distinct object about which data is saved. Entities are an essential part of the entity-relationship model and therefore need to exist and to be of the correct type and contain the correct data and formats.

Our third category of learning topics is the Cartesian product. In this category learners have to understand problems of how to combine different entities taken from two sets of the database to create a product from them.
When designing the learning tasks, we found that in some task descriptions the very entities connected by the relation in question were mentioned while they were not in other tasks. We thus had two kinds of tasks. While learners might very well be able to identify the correct cardinality of a relation between two entities when they are directly connected, they might have problems to identify a problem when the entities are indirectly connected via a third entity or even a number of entities. The same may also happen in tasks from both of the other categories.

V. MULTI CATEGORY TASK CLASSIFICATION

As mentioned earlier, similar tasks are aggregated into categories to provide the spaced repetition algorithm with a way to present the learner with a task that represents the concept scheduled for a certain point in time while avoiding repeated presentations of the very same task. In a flashcard application presenting the same word repeatedly is unavoidable and actually fosters knowledge. In a learning game about database where concepts are concerned, the learner should tackle a problem dealing with “1-N” relations but he or she should not be bored by having to solve the same problem over and over.

This way, classifying task items as belonging to categories can make the game more engaging. Task classification does, however, also come with a downside. Often, tasks can be described with a number of orthogonal categories. For instance in some task from the ER domain, the task descriptions mention the very entities connected by the relation in question, while they were not in other tasks. We thus had two kinds of tasks even if both were dealing with the same cardinalities.

In a first approach we currently deal with this circumstance by seeing the combination of all orthogonal categories describing one item as one category. So currently there are categories for tasks with directly connected entities for each cardinality and categories with indirectly connected entities for each cardinality. While this approach allows working with multi category task descriptions, it does not make use of the fact that some dimensions (like cardinality) are crucial for the description of the learning content while others (like direct-indirect) are merely descriptions of the type of task. Task descriptions might, however, also be useful for the selection of learning content as they convey information about the task’s difficulty.

In future versions, we plan to handle task descriptions separately from content description categories. Then spaced repetition will be solely influenced by learning content descriptions. Task descriptions can be used in a second step to select which task in the content category selected by spaced repetition will be selected. Task selection within a content category can then be based on properties of the task description. For instance more difficult tasks like indirect problems in the ER domain can be presented to the user at later stages of the game to make game play more challenging.

VI. IMPLEMENTATION

As already mentioned the idea of the app is to create an interesting and playful way of learning. The user should be able to understand and handle the app without or only a short introduction, so it is a self-explanatory app. To create a challenging experience and to motivate the user, the app will save and analyze which kind of mistakes the user is able to handle and choose the complexity factor of the stories accordingly.

Our concept of a database learning game aims to foster the understanding of the entity-relationship model. In order to do so, there are different stories which are represented as an ER model. The task for the user is to find the mistake(s), correct and in some cases to identify it. One can find different kinds of problems and due to these variable solutions.

At the beginning of each exercise the app presents the story and its representation as an ER model to the learner. Within this representation there are for example one or more logical errors with the cardinality constraints which have to be corrected by the learner. In order to do so he has to tap on the respective cardinalities which he expects to be wrong. He then gets shown a menu from where he can pick the supposed right cardinality to replace the wrong one. When the learner believes that he has found all wrong cardinalities and replaced them with the right ones, he can tap on a button to see if the was correct. There are two implementations of this button. One just gives the learner an immediate feedback and informs him about his success. The other also shows him the positions of the incorrect cardinality constraints. An example for this can be seen in Figure 2.

![Figure 2: Cardinality selection in prototype learning app](image)
An example for a story for a task from the cardinality category would be the following:

“Mr. Mayer is an English teacher at a high school. Because a colleague from the administration department became ill for a longer period, Mr. Mayer has to take over her tasks. For this time, he is relieved from teaching. Nevertheless, there are always lessons on the schedule for Mr. Mayer. Please find and correct the mistake.”

The story concept is also used with the other categories. In tasks from the entity-category, the user has to create a new entity, place it on the correct position and to find the asked for relationship. In tasks from the Cartesian product category, there are always problems of how to combine different entities taken from two sets. With the help of a short story the user should be able to try different combinations or only to choose the correct quantity from a pool of proposals.

Based on the answers, a score is saved for the respective category, indicating in which categories the learner already has a proper knowledge and in which areas he still needs to improve. This is done separately for both used algorithms. Based on this score and the other values used by the algorithms, the game then chooses the category to be presented next and the optimum interval for the next repetition of the different categories.

After the first session of each round of play, the performance of the learner is fed back into the SM2 algorithm as its quality of the response. When the game is launched for the first time, the algorithm in the background begins with some default values for its variables. According to the SM2 algorithm the first two repetition intervals are fixed. After this initial phase the algorithm then schedules the subsequent repetitions based on the learner’s performance for each category.

VII. CONTENT SELECTION ALGORITHMS

As mentioned earlier, the algorithm behind the scheduling of our database game relies on the SM2 algorithm. It was developed by P.A. Wozniak in 1997 and uses different variables to calculate the most appropriate time for the next repetition. This algorithm has proven its reliability especially when being used in scenarios like learning with flashcards. However, using this algorithm alone in learning games is not always appropriate. Learning games are aimed at keeping the motivation of a learner high and thus stay with the game for a certain time. This should lead the learner to play the games several games in a row.

We have therefore developed another algorithm (FS algorithm) which is a round-based algorithm in contrast to the time-based SM2 approach. Just like with the SM2 algorithm the FS algorithm increments a score value (which is equivalent to the quality of answer in SM2) on right answers and decrements the score on wrong answers. Additionally there is another value called relevance which is increased for the current item by 1.5 on right answers and by 1 on wrong answers. All other categories get decreased by 0.2. The sum of score and relevance is called the rank. The rank then determines the content to play in the next round if the player should chose to carry on playing. Additionally a flag is set for the last played content. Therefore the unflagged item with the lowest rank is to be played next. The algorithm calculates a ranking which sorts the learning topics by how well the learner has answered them after each round of play. This data is kept separate from the data used by the SM2 algorithm in order to ensure that the scheduling of the repetition does not get corrupted but at the same time offering a similar learning experience as with the SM2 approach. This ensures that the learner gets presented the content that he or she is the least familiar with and which was not played in the last round.
However, each time a new round of playing the game is started, the SM2 algorithm makes the first decision about the content selection. Only when the user decides to play the game several games in a row, the FS algorithm takes over the decision-making. Therefore, both, the SM2 algorithm and the FS algorithm work together in our prototype app as can be seen in Figure 3.

The categories are kept separately within the game. Each learning topic category is stored with its values that are used by the algorithms. After the algorithm in charge has determined the category to be learned next, a task from this category is selected and presented to the learner. After he has completed the task, the algorithm schedules the next repetition based on his performance.

VIII. CONCLUSION AND FUTURE WORK

Categorizing learning content can help to focus more on concepts rather than on bare facts. Our implementation has shown that content in learning games may very well be categorized in order to schedule repetitions of the respective content by certain concepts. However, this categorization and dealing with it can still be optimized. This is especially necessary when content of the same category and also with multi categories becomes more complex. While learners might be able to learn the concept of 1:n cardinalities in an ER diagram very quickly when they just have to consider one relation, it might be much harder when they have to consider more than one relation or even have to change one relation to make one at a different position correct. In the current implementation the category would just be “1:n”, no matter how difficult the occurrence of the respective task might be. This may not provide the best possible insight on how well a learner knows about that concept and thus not the best possible scheduling of repetitions.

Another problem can occur if a user repeatedly only plays one game in a row, but continuously logs off and on again afterwards. This will always trigger the SM2 algorithm in a very short amount of time and therefore corrupt the spaced repetition approach.

In future work we will therefore not only enhance the game’s intelligence dealing with this behavior but also consider the level of difficulty of each implementation of the respective category. This might lead to a further classification of a category. For example there may be easy, medium or hard occurrences of the “1:n” category, determined for example by whether the respective cardinalities are directly or indirectly connected. Based on that there will also be a further enhancement to how the used algorithms will deal with those categories.

Using the two algorithms, this would supposedly mean that the categories need to get identified as one of the occurrences and their difficulty. This might be done by comparing the learner’s performance with the times the respective category was presented to him. From this we might be able to draw a conclusion about how difficult this category is to learn for him. The algorithm then needs to determine not only the category and select a content item from it but also its occurrence. One way to deal with this problem might be to present the different occurrences in a stage-based manner, starting with supposedly easy tasks and then raising the difficulty level subsequently, which would also make game play more challenging over time.

Regarding multi category content selection there might also be task descriptions containing different (i.e. orthogonal) categories, which would be an even more advanced approach. In order to accomplish an optimum content selection and optimum repetition intervals, there would have to be several adjustments to be made with the stored meta-data to enable the algorithms to make the best possible decisions. One way to deal with it could be to choose a two-step approach which first considers the occurrence of a learning task and then the difficulty or the other way round.

REFERENCES