Learn and Evolve the Domain Model in Intelligent Tutoring Systems

Approach Based on Interaction Traces

Aarij Mahmood Hussaan\textsuperscript{1} and Karim Sehaba\textsuperscript{2}

\textsuperscript{1}IQRA University, Defence View, Karachi, Pakistan
\textsuperscript{2}Université de Lyon, CNRS, Université Lyon 2, LIRIS, UMR5205, F-69676, Lyon, France

aarijhussaan@iqra.edu.pk, karim.sehaba@liris.cnrs.fr

Keywords: Adaptive Learning System, Bayesian Student Model, Clustering, Interaction Traces.

Abstract: Majority of the systems developed in the field of education aim to tailor information presented to their users according to them as efficiently as possible. Often these systems use a User Model, a Domain Model and an Interaction Model to provide the adaptation/personalization effect. However, any deficiencies in these models directly influence the quality of adaptation the system provides. In order to address this issue, we propose to use modeled user’s interaction traces to detect these deficiencies in the domain model. Furthermore, the use of these interaction traces will help us in proposing a more correct domain model according to the user’s competence. We had tested our approach by creating an educational system. The overlay modeling approach is used with Bayesian Networks to model our domain model. The results of the conducted experiments are also presented in this article.

1 INTRODUCTION

Adaptive computer based educational systems aim to provide personalized education to their users. The quality of the adaptation depends directly upon the quality of the educational domain’s knowledge modelling in the system. Since, knowledge modelling is not a straight forward task, therefore it is sometimes necessary to update the system’s knowledge about the educational domain. This updating of knowledge can be performed manually (by a human expert), automatically (by machine learning), or semi-automatically (using machine learning to help human expert).

Our research focuses on the problem of semi-automatically finding improvements in the Domain Model\textsuperscript{1} of a computer based educational system via data-mining, and proposing these improvements to the system’s domain expert\textsuperscript{2}.

Data-mining is the process of extracting information from a raw dataset and transforming it to knowledge. Researchers have made use of data-mining in various educational settings for different objectives (Bhise et al., 2013). (Baker, 2010) have identified four key areas data-mining application in education, they are: improving student models, improving domain models, studying the pedagogical support provided by learning software, and scientific research into learning and learners.

In our research we focus on the type of educational systems generally referred to as Intelligent Tutoring System (ITS) (Burns, H. L. And Capps, 1988) or Adaptive Hypermedia Systems (AHS) (Brusilovsky, 1996) in literature. Many of these systems follow the architecture proposed by (Benyon and Murray, 1993). They have identified the most important components of an ITS, they are

1. User Model: represents the system’s belief about the user, particularly: psychological model, profile model and student model.
2. Domain Model: defines the aspects of the system and the world that are important for inferences, e.g., knowledge about the domain, teaching resources, functions that might be altered.
3. Interaction Model: handles the dialogue between the user and the application for adaptation of the system to the user properties taking into account the domain model.

\textsuperscript{1}In this article, domain model refers to the description of the various concepts of an educational domain and the relation between them. For example, for the educational domain of Mathematics the concepts could be ‘Addition’, ‘Subtraction’, etc.

\textsuperscript{2}A domain expert is a person who defines the education domain model of an educational system.
In our research, we propose to represent the users’ interaction history as modelled interaction traces (Clauzel et al., 2011). The interaction traces are defined as a history of learner’s actions collected, in real time, from his/her interaction with a computer system. We have applied the data-mining process to these modelled interaction traces to discover new knowledge about the domain model. Afterwards, this knowledge is presented to the domain expert, who can make “if” necessary adjustments/modifications to the domain model. These adjustments can have a positive effect on the educational quality of the system. These improvements could include: identifying the exercises or educational resources that are not related to the correct concept, and splitting a concept into further new concepts.

To test our approach we have created an adaptive computer based education system. We have used overlay modelling using Bayesian Network (BN) (Pearl, 1988) to model our user and domain model. The use of BNs to manage uncertainty in user models has been investigated and championed by (Conati et al., 2002). In our system, the user model is updated by the propagation of values in the BN.

Many approaches have been proposed to update the domain model using the users’ logs. (Jr et al., 2009) used learning factor transfers and Q-matrices to generate domain models that maximizes learning via item sequencing. Similarly, BNs have been used extensively to model student/user models, (Sande, 2013) used BNs to trace student performance. Clustering techniques have been also employed to group students into different categories, (Xu et al., 2013) compared the efficiency two clustering techniques in educational systems. (Shen et al., 2003) used clustering and case-based reasoning to personalize the learning process of students. (Retalis and Papasalouros, 2006) proposed a tool that employs clustering to help instructors to learner’s progress and make amendments to the teaching strategy if necessary. Clustering with other techniques have been employed by (Chen and Chen, 2009) to provide formative assessment of students to teachers. Domain concept similarity was measured in (MADHYASTHA and HUNT, 2009) using mining techniques on students’ assessment data. Along with these many other approaches have been employed to either discover new concepts, and or cluster students (Lee et al., 2009).

Although, many systems try to update the domain model in different ways. However, none of them use modelled interaction traces as knowledge sources for the data-mining process. Our approach, unlike others, models the user’s interaction traces, and use them to update the domain model, while also using the same traces to update the user profile.

The remainder of the article is organized as follows. Section 2 gives the general architecture of our system. Section 3 gives the details about the representation of the domain knowledge and user profile. Section 4 shows the updating process of domain knowledge and user profile. Section 5 shows the simulation we conducted with our approach and results we obtained. We finish by our concluding remarks and perspectives.

2 GENERAL ARCHITECTURE

In order to bring flexibility and adaptability in supporting individualized learning, the system must follow the progress made by each user in order to propose relevant information to his needs and skills. For this, the proposed architecture includes control loops, both in on-line for the adaptation of the scenario based on user’s behaviour, and in off-line for updating domain knowledge and user profile based on the interaction traces.

Our system’s architecture is depicted in the Fig 1. It has six modules: the domain model/knowledge, the user profile, the reasoning, the interactions management, the interaction traces base and the traces management module.

![Figure 1: General architecture.](image)

Initially, the domain expert feeds the system with domain knowledge about an educational domain and characterizes the user profile (in some application context, the user profile can be defined by the user himself). The reasoning module uses the user profile and domain model, in real-time, to generate learning scenarios adapted to the user and his educational goals. We define a learning scenario as an ordered sequence of learning activities and exercises that helps the user to achieve an educational goal.

During the interaction between the user and the scenario, all the user actions are collected by the in-
teractions management module, where they are modelled as modelled interaction traces. The traces management module use them, in off-line mode, to update the domain knowledge and the user profile.

The following paragraphs give a short overview of each module:

**Domain Model/Knowledge**: contains all characteristics of the knowledge to teach an educational domain. It consists in storing information on topics/concepts, exercises, problems and relationships between them. This knowledge, represented as an extended Bayesian network, is used by the reasoning module for the generation of learning scenarios. The Sec 3 gives more details on the representation of knowledge.

**User Profile/Model**: represents and manages all important information about the user/student, which can aide in the learning process. All these information are represented as a set of pairs <attribute, value>, where value can be discrete or probabilistic. For example, <age, 23>, <Math, 70.0>, etc.

**Reasoning**: the purpose of this module is two-fold. Firstly, to generate adaptive scenarios according to the user's profile and domain knowledge via The Recommender sub-module. Secondly, to tailor the presentation of the scenarios selected by the recommender according to the user’s profile via the The Presentation Adapter sub-module.

**Interactions Management**: is responsible to log all the user’s actions (responses to questions, response time, time consumed while consulting help, type of device used by the user to interact with the system etc.) in a modelled trace (see Sec 4.1). The Traces Management module access them in order to learn new knowledge about the user.

**Interaction Traces Base**: this is where the users’ modelled interaction traces are stored.

**Traces Management**: is responsible to update the domain knowledge and user profile using the users’ traces. The reasoning performed by this module takes place off-line i.e. the User Profile and Domain knowledge are updated only when the user logs out. The section 4.1 gives the traces representation and the reasoning process adopted by this module.

In the next section we show the modelling of the domain knowledge and user profile.

### 3 REPRESENTATION OF DOMAIN KNOWLEDGE AND USER PROFILE

The domain knowledge is modelled via a domain model. The domain model defines the concepts needed to describe the educational domain. The topics/concepts in the domain model are interconnected to form a hierarchy. We have used the overlay Modelling approach to model the domain model. Overlay models need association with a statistical model to infer user’s knowledge (Nguyen and Do, 2009). Therefore, we used Bayesian Networks (BN), they are the best choice as shown by (Brusilovsky and Millán, 2007), because they allows to represent and infer the uncertainty in the user’s knowledge.

Our domain model is an extended Bayesian Network (BN) whose nodes represent the concepts to teach and links represent hierarchical relationships between the concepts. Wherever necessary the general concepts are further divided into more specific sub-concepts. The conditional probabilities of the BN are defined by the domain expert\(^3\). Figure 2 shows an example a domain model, in which:

1. Concepts of the domain, represented by circles on a white background, represent the atomic elements of knowledge about given topics. In the knowledge tree, each concept is divided into sub-concepts, and these into other sub-concepts, and so on. At the same time, each concept of the network contains sets of definitions and examples related to the topic. These elements are in several formats (text, image, video, etc.).

2. Evidences or observation, represented by circles on a grey background, are the leaves of the tree. The observations are exercises and problems related to the concepts which they are linked.

\(^3\)The conditional probabilities can also be learned automatically via different techniques but they are beyond the scope of our research.
exercises and problems are used for the creation of the tests.

Indeed, we distinguish between problems and exercises as shown in several studies in the learning field. An exercise can be defined as an activity that is performed as a test or practice of one’s technical skill. If the students know how to perform and know the steps involved in reaching a solution, then they are performing an exercise. It is noted that the exercises can be hard (Singh and Lau, 2006), but they are never puzzling, for it is always immediately clear how to proceed and solve a problem algorithmically by recognition, recall and reproduction. Problems are something completely different. It is a question that, on the one hand, concerns several concepts in relation (Bair et al., 2000), on the other hand, the solution requires a procedure not immediately apparent. It consists in putting the student in an original situation that requires him to put together several concepts to find new results for him.

We therefore distinguish two types of evidence variables, namely:

1. Exercises: these are evidences associated with simple concepts (i.e. not containing sub-concepts). For example, the evidences of the concept “Add” concern exercises of addition concept.

2. Problems: these are evidences associated with concepts containing sub-concepts. The evidence in this case must concerns the concept in question and all its sub-concepts. For example, the evidences of the concept “Arithmetic Operations” are problems that deal with both addition and multiplication.

Since we are using an overlay approach, the user model contains the probabilities of the user’s mastery of different concepts in the domain model. For example, a particular user model may contain a value “<Math, 75.0>”, indicating that the probability that the user knows the concept “Math” is 75%.

The updating of the domain model consists of adding or deleting the concepts and/or the relations between the concepts in the BN, and the updating of the user profile consists of calculating the probabilities of the variables of the network according to the observations. Recall that the user profile is represented by a vector of components of the form <attribute, value> where the attributes corresponds to the variables of the network (concepts of domain knowledge), and the values correspond to the probabilities of the variables of the network (estimation of the user’s mastery of concepts).

The next section presents the strategy we have adopted to update the domain knowledge and the user profile.

4 UPDATING OF DOMAIN KNOWLEDGE AND USER PROFILE

The Fig.3 gives the principle of our approach for updating the domain knowledge and the user profile. Initially, the expert defines a domain knowledge in the form of an Extended Bayesian Network presented in the section 3. Thereafter, the reasoning module uses this knowledge to generate leaning scenarios to the user taking into account his profile. In the learning session, all the actions of the user on each scenario tests are stored in the form of modelled interaction traces. So, each user has a trace base representing all his interactions in different sessions. Our strategy consists in using these traces as knowledge source in order to update both the domain knowledge and the user profile.

Figure 3: Domain Knowledge’s and User Profile’s updating process.

To update the user profile, the idea consists in using Baye’s formula (cf Sec 4.2) for the propagation of information in the network. For the updating of domain knowledge, the principle is to categorize the observed elements of the trace using a method of clustering. The results of the clustering are then presented to the expert who can then update the structure of the Domain Model according to the results.

Firstly, this section presents the model adopted for the formalization of traces of interaction. Secondly, we present the strategy for updating the user profile. Finally, we detail the method of updating domain knowledge.

4.1 Formalization of Traces

A trace is the result of the interaction between the user actions and the scenario tests, where the scenario
tests are created by the reasoning module. Each session is associated to a trace. A trace is a sequence of observed elements representing the user actions on the elements of tests scenario (exercises and/or problems). Formally, a trace \( T \) is represented as follows: \( T = < O_1, O_2, ..., O_n > \) Where each observed element \( O_i \) is characterized by the following quintuple: \( O_i = \langle Q, R, T, S, E \rangle \)

\[
Q: \text{Exercise / Problem (Evidence/Observation node of BN)} \\
R: \text{Response of user} \\
T: \text{Response time : time elapsed during the display of the question and the response of the user.} \\
S: \text{Evaluation of user response.} \\
E: \text{A function that calculate the gap between the user response and the correct response. This function takes into account also the total time that he/she spent to answer the question} Q.\text{ The values of this function } f \text{ are defined in the interval } [0,1] \text{ where: } 0 \text{ indicates that the student does not know the concept, and } 1 \text{ indicates that the concept has been correctly learned. In order to simplify, we use this classification:} \\
1. \text{If } 0.1 \geq f \geq 0 \text{ then } E = \text{Very Bad} \\
2. \text{If } 0.3 \geq f \geq 0.1 \text{ then } E = \text{Bad} \\
3. \text{If } 0.6 \geq f \geq 0.3 \text{ then } E = \text{Average} \\
4. \text{If } 0.8 \geq f \geq 0.6 \text{ then } E = \text{Good} \\
5. \text{If } 1 \geq f \geq 0.8 \text{ then } E = \text{Very Good} \\

\]

### 4.2 Updating User Profile

The probabilities associated with each concept of the BN represent the mastery achieved by the user of that concept. The response of the user i.e. a correct response of an exercise of a concept given in relatively quick time, with little or no help will increase the probability of the user knowing that particular concept. Similarly, an incorrect response will decrease the probability of the user knowing that particular concept. This updating of the probabilities from the evaluation of user’s responses is done by the use of Baye’s Rule.

Let \( A = \{ A_1, A_2, ..., A_i \} \) be the set of observations related to the topic B then:\(^4\)

\[
P(A|B) = \frac{P(B|A) \cdot P(A)}{\sum P(B|A_i) \cdot P(A_i)} \tag{1}
\]

\(^4\)Note: The same formula is used in the propagation of probabilities between concepts.

### 4.3 Updating Domain Knowledge

Recall that the principle of updating domain knowledge is to semi-automatically find new concepts and links in the domain model, which can help in increasing the quality of the domain model. This in terms of BN means learning or finding new hidden nodes in the network. For BNs there exists many techniques to facilitate hidden nodes learning (Neapolitan, 2003). These techniques are primarily based upon statistical methods. Unfortunately, for us these techniques are not feasible as they do not perform well on small sized data set. And we don’t know a priori how large the data set will be. To counter this problem, we found out that Clustering Algorithms are more appropriate for our purpose. Since they can perform reasonably on small-sized data-set as well. It consists in using these algorithms on the observed elements of the trace in order to identify new concepts and relationships that emerge. That is to cluster the observations according to the value of the function \( f \).

There are many algorithms to do clustering\(^5\). We experimented with K-Means algorithm and found it to be adequate enough to test and validate our approach. To perform clustering, we follow the following steps:

1. Preparing the Data: Make the data compatible to be presented to the clustering algorithm. We present the data in the form of a matrix. Where, the rows are observations and columns are the traces. If we take Fig.3 for reference, then the matrix will be like.

<table>
<thead>
<tr>
<th>Evidences</th>
<th>Trace 1</th>
<th>Trace 2</th>
<th>Trace N</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 + 2</td>
<td>Good</td>
<td>Very Good</td>
<td>–</td>
</tr>
<tr>
<td>15 + 56</td>
<td>–</td>
<td>Very Good</td>
<td>Good</td>
</tr>
<tr>
<td>4.2 + 5.6</td>
<td>Bad</td>
<td>Very Bad</td>
<td>Bad</td>
</tr>
<tr>
<td>1 + 4.3*5</td>
<td>–</td>
<td>–</td>
<td>Average</td>
</tr>
</tbody>
</table>

\(^5\)A detailed comparative study of different clustering algorithms is out of the scope of our research.
different clusters. These clusters represent the new concepts in the Domain Model. These clusters show how the domain model could be organized according to a particular user. Once we have the clusters, we present the findings to the domain expert, who can validate the results and update the domain model accordingly.

5 SIMULATION AND RESULTS

To validate our approach we performed experiments. To show the generic nature of our approach we have decided to conduct our experiments for a course of JAVA programming language. The domain model of this course is constructed by consulting the tutorial provided by Sun Microsystems. This tutorial is publicly available on internet. Since, we didn’t have access to real-world classrooms; we decided to use simulated users for our experiments. The use of simulated students have been done in numerous studies. (Iglesias et al., 2008) used simulated student clusters to simulate student behaviour.

The original concept tree for the tutorial of JAVA programming language is shown below in Fig.4.

![Figure 4: Actual Model of Java Tutorial.](image)

There are plenty of concepts in this concept tree. In this article we have only concentrated on the concept ‘Classes & Objects’. Though, in the original tutorial the concept ‘Classes & Objects’ is well classified into sub-concepts such as interface, inner classes etc. For our experimentation we deliberately haven’t classified the concept ‘Classes & Objects’ i.e. more than one sub-concept of ‘Classes & Objects’ is merged into a single concept ‘Classes’ in our domain model. The domain model represented in the form of Bayesian Network of our system with the modifications is shown in Fig.5). This is done deliberately to replicate the behaviour of an expert who either didn’t consider it important for the course or had made an error in the classification. This means that the scenarization module will consider all the exercises/problems of the concept ‘Classes’ of the same difficulty, which is not the case. The sub-concepts of ‘Classes’ varies in difficulty for e.g the sub-concept ‘Object’ is much easier than the subconcept ‘Interface & Polymorphism’. The assumption made by the scenarization module will cause problem for some users( not having sufficient knowledge) as the exercises/problems will not be presented to them in a progressive manner. This will result in either loss of the user’s interest or inhibited learning.

For our experiments, we have decided to create three profiles of users. Each profile will have its own characteristics and competencies. These profiles are created to model the real world, where often users in the same group have different competencies and characteristics. Since, here we are dealing with a programming course for JAVA language; and in the paradigm of JAVA we are dealing specifically with the concept ‘Classes’. These profiles are created keeping in mind the different competencies of the users in JAVA. The characteristics of each profile are shown below.

![Figure 5: Modified Model of Java Tutorial.](image)

1. **Profile 1**: basic users of JAVA language; they know the language fundamentals of JAVA.
2. **Profile 2**: knows the language fundamentals and also the basics of the concept ‘Classes’ of JAVA but not the advanced topics of ‘Classes’ like inner classes, Annotations etc.
3. **Profile 3**: are advanced users with background knowledge of a high-level language. They also

---

6 Although, we have chosen JAVA but we could have chosen any educational domain that can be represented via concepts and relations/links between those concepts. For example, Mathematics, Physics, Chemistry, English Language, etc.
7 @ http://java.sun.com/docs/books/tutorial/.
8 In this figure only partial tree is shown.
9 Though, we are concentrating on only one concept, the process remains the same irrespective of the number of concepts involved in the process.
10 These profiles are constructed after consultation with a domain expert in JAVA. Different profiles may have resulted in different clusters, however, the process would have remained the same.
knows the advanced sub-concepts of the concept 'Classes'.

After the profiles are created, we defined the scenarios for the course. The scenarios, as mentioned before, can be either created by an expert or they can be automatically created by the scenarioization module. For our study we had defined the scenarios manually. We had selected 25 exercises. These exercises covers 4 sub-concepts of the concept 'Classes'. The distribution of the exercises is shown below:

1. Question 1 - 6: belongs to the type 'Basic Questions'
2. Question 7 - 14: belongs to the type 'Inheritance'
3. Question 15 - 20: belongs to the type 'Interface and Polymorphism'
4. Question 21 - 25: belongs to the type 'Inner Classes'

We had created three scenarios for the concept and each scenario contains 10 exercises. Each scenario contains exercises from all four sub-concepts.

After the creation of scenarios, we presented them to the users of each profile. As a result of the interaction between the user and the scenarios traces are generated. These traces are saved in our trace base module. Afterwards, we performed clustering on the traces to discover knowledge. As mentioned earlier, before performing clustering the traces needed to be transformed in the form of matrix. The matrices for the three profiles after transforming and handling missing values are shown below:

<table>
<thead>
<tr>
<th>Profile 1</th>
<th>Observations</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Good</td>
<td>Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td></td>
</tr>
<tr>
<td>11 Very Bad</td>
<td>Bad</td>
<td>Very Bad</td>
<td>Very Bad</td>
<td></td>
</tr>
<tr>
<td>19 Very Bad</td>
<td>Very Bad</td>
<td>Very Bad</td>
<td>Very Bad</td>
<td></td>
</tr>
<tr>
<td>25 Very Bad</td>
<td>Very Bad</td>
<td>Very Bad</td>
<td>Very Bad</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile 2</th>
<th>Observations</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td></td>
</tr>
<tr>
<td>11 Good</td>
<td>Very Good</td>
<td>Good</td>
<td>Good</td>
<td></td>
</tr>
<tr>
<td>19 Average</td>
<td>Average</td>
<td>Average</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td>25 Bad</td>
<td>Bad</td>
<td>Bad</td>
<td>Bad</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profile 3</th>
<th>Observations</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td></td>
</tr>
<tr>
<td>11 Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td></td>
</tr>
<tr>
<td>19 Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td></td>
</tr>
<tr>
<td>25 Average</td>
<td>Average</td>
<td>Average</td>
<td>Average</td>
<td></td>
</tr>
</tbody>
</table>

Clustering was performed after transforming the traces. The results of clustering are shown below:

<table>
<thead>
<tr>
<th>Profile 1</th>
<th>Cluster No</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Questions belonging to category 'Basic Questions'. Ex Question 1, 11</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>All other questions. Ex Question 11, 19, 25</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6 shows the final configuration of the domain model for each of the three profiles. In the figure, for Profile 1 the clustering process have made a separate cluster for the 'Basic Questions' and the rest of the questions are clustered separately. This goes well with our initial hypothesis that a user of basic qualification (Profile 1) will not be able to solve the questions of advanced concepts. These clustering results might prompt the expert to break the original concept of 'Classes' into two different 'Sub-Concepts' of 'Basic Concept' and 'Advanced Concepts'.

Similarly, if the domain expert analyse all the results of different profiles, s/he might be able to come up with a domain model that will further enhance the learning experience of users.

Figure 6: The resulting structure of the original Domain Model of the three profiles after clustering.

### 6 CONCLUSION

In this paper, we presented an approach to semi-automatically update the domain model using modelled interaction traces as knowledge sources in computer based educational systems.

We have used clustering to search for new knowledge in the domain model, afterwards, we present
this knowledge to the domain expert who can then decide to incorporate this knowledge in the domain model or not. The quality of clustering results depends upon the different parameters of the clustering algorithm. To help the expert with different clustering algorithms, we have allowed the expert to select different parameters via an easy-to-use GUI in our system\textsuperscript{11}. However, the choice of clustering algorithms or different parameters do not effect on the general working of the system/approach.

We also tested our approach with simulated users. The results obtained showed that our work is valid. Although, we acknowledge that different profiles could have resulted in different results. However, our approach would not have been changed as a result.

We have also published papers about the scenario-ization module. This module adaptively selects learning activities for a user according to his educational goals and user profile. We will definitely test our approach with real-users. This will further cement the scientific standing of our approach.

REFERENCES


Clauzel, D., Sehaba, K., and Prié, Y. (2011). Enhancing synchronous collaboration by using interactive visu-


\textsuperscript{11}The implemented system has not been presented in this article.