An Adaptive Tutor to Promote Learners' Skills Acquisition during Procedural Learning

Joanna Taoum, Anaïs Raison, Elisabetta Bevacqua and Ronan Querrec

Lab-STICC, UMR 6285, CNRS, ENIB, France
{taoum,raison,bevacqua,querrec}@enib.fr

Abstract. Our research work proposes an adaptive and embodied virtual tutor based on intelligent tutoring systems. The domain model is represented in our work by a virtual environment meta-model and the interface by an embodied conversational agent. Our main contribution concerns the tutor model, that is able to adapt the execution of a pedagogical scenario according to the learner's level of knowledge. To achieve such a goal, we rely on the inference of the learner's memory content.

Keywords: Adaptive Pedagogical Behavior · Virtual Environment · Learner's Memory · Pedagogical Scenario · Embodied Conversational Agent

1 Introduction

The work presented in this paper is applied to the domain of procedural learning in a virtual environment for industrial systems. According to Anderson [1], procedural learning is considered to be complex and this complexity requires the use of practice (repetition). In order to be able to manage the interaction between a tutor and a learner during these repetitions, we choose to describe this information using pedagogical scenarios. These scenarios define the activities that should be carried out by the tutor and the learner, their sequencing, as well as the pedagogical objectives that should be achieved.

However, these scenarios remain general. They can be effective at the beginning of learning (during the first repetitions), but not in the following repetitions. Considering that each learner evolves differently, during repetitions, it is important to adapt the execution of these pedagogical scenarios according to the learner's evolution.

The real-time adaptation of the pedagogical situation to a learner is one of the major objectives of Intelligent Tutoring Systems (ITSs). In order to adapt the situation to the learner, a fundamental goal of an ITS is to model the learner. In procedural learning domain, Corbett and Anderson [2] propose some general concepts to model the learner during the acquisition of procedural skills. These concepts are too theoretical to be applied to teaching procedures in industrial systems. As we are dealing with teaching human activities in industrial systems, the cognitive knowledge that our student model infers is related to memorization. Atkinson and Shiffrin [3] proposed a general theoretical framework which divides human memory into three structural components: sensory memory, working memory and long-term memory. To implement this general framework of memory, several ITSs have been built using the cognitive architecture Act-r [4]. The goal of Act-r is to simulate the realization of complex tasks by human beings. It is mainly designed around two concepts: declarative and procedural knowledge. Declarative knowledge is represented by a set of chunks and procedural knowledge by a set of production rules (*if-then* statements). In Act-r, information processing of memory is a *Black Box*. It can be used to generate the tutor behavior but not to represent the knowledge flow in the learner model.

In this work, we propose a tutor behavior that adapts the execution of the pedagogical scenario according to the learner's inferred knowledge (see section 3.1). To represent such a knowledge, we propose a cognitive architecture based on Act-r [4]. In section 2, we introduce Mascaret [5] that we use to represent the domain model and the pedagogical scenario. To realize pedagogical assistances in a human-like way, we propose an interface model based on a virtual environment and an Embodied Conversational Agent (ECA).

2 Domain and Interface Model

The domain model is formalized in our work by Mascaret, a virtual reality meta-model based on UML. It allows to describe and simulate technical systems and human activities in a virtual environment. The domain expert uses class diagrams to describe the different types of entities, their properties and the structure of the environment. Procedures are designed as predefined collaborative scenarios through UML activity diagrams, which represent plans of actions. It is the role of the interface model to recognize when the student executes these actions. Using a meta-model to formalize the domain model 1) allows domain experts to provide the knowledge themselves in the ITS, and 2) keeps domain data explicit during the simulation, thus they can serve agents as the knowledge base.

In Mascaret, pedagogy is considered as a specific domain model. Pedagogical scenarios are implemented through UML activity diagrams containing a sequence of actions. These actions can be either pedagogical actions, like explaining a resource, or domain actions, like manipulating an object. For the definition of pedagogical scenarios and actions, we rely [6]. In Mascaret five types of pedagogical actions are implemented:

1. Pedagogical actions on the virtual environment: highlighting an object, playing an animation.
2. Pedagogical actions on user's interactions: changing the viewpoint, locking the position, letting the student navigate.
3. Pedagogical actions on the structure of the system: describing the structure, displaying a documentation about an entity.
4. Pedagogical actions on the system dynamics: explaining the procedure's objectives, explaining an action.

5. Pedagogical actions on the pedagogical scenario: displaying a pedagogical resource, making an evaluation (e.g. a quiz).

These pedagogical actions are realized through the interface model, that is represented in our work by an ECA, using Greta platform [7]. This ECA is able to select and perform multi-modal communicative and expressive behaviors in order to interact naturally with the user. In Mascaret, any entity which acts on the environment is considered as an *agent*. Particularly, the ECA and the human user are *embodied* agents. An embodied agent is able to recognize as well as perform basic actions, like:

1. Verbal communication (e.g. giving an information)
2. Non-verbal actions (e.g. facial expression) and actions on the environment (e.g. manipulating an object)
3. Navigation (e.g. observing)

These basic actions are used to implement the domain and pedagogical actions involved the pedagogical scenario. Through the interface model, the tutor is able to recognize the realization of each of these actions performed by the user to evaluate the evolution of the pedagogical scenario and to adapt it if necessary.

3 Adaptive Tutor Model

The tutor model uses the knowledge of the domain model and the actions done by the learner in order to choose pedagogical actions that will be realized through the interface model. More precisely, the tutor behavior takes into account the actions done (or inaction) by the student by recognizing them through the interface. The goal of our proposed tutor model is to adapt the execution of the pedagogical scenario according to the student model represented in our work by the student's memory.

In what follows, we first describe the student model that is used to decide which adaptation to perform and then how the tutor behavior detects the need for adaptation.

3.1 Student Model

We propose a reimplementation of the generic framework of memory proposed by Atkinson and Shiffrin [3] in the context of learning procedures. Our contributions to this framework consist in making explicit the *Black Box* by 1) formalizing the user's memory information, and 2) implementing the transformation of the stimuli into knowledge and the knowledge flow between the three components of the human memory. In our work, incoming stimuli from the virtual environment and the virtual tutor are restricted to those related to vision and hearing. Thus, the student can see 3D objects and hear instructions uttered by the tutor about activities to realize. Therefore, we encode data about objects and activities. To formalize the encoding of information, we rely on Mascaret.

Objects are considered in MASCARET as Entity. An Entity can be hierarchical, thus it can be composed of Entity and represented by a name, geometric properties (position, orientation and shape) and domain model properties (as a meta class Class attribute). As for activities, they are represented by the meta class Activity, they can also be hierarchical and composed of several Activity, Role, Action and Flow between actions and objects. MASCARET data formalism is hierarchical, which allows to instantiate the content of the memories according to the knowledge level of the learner.

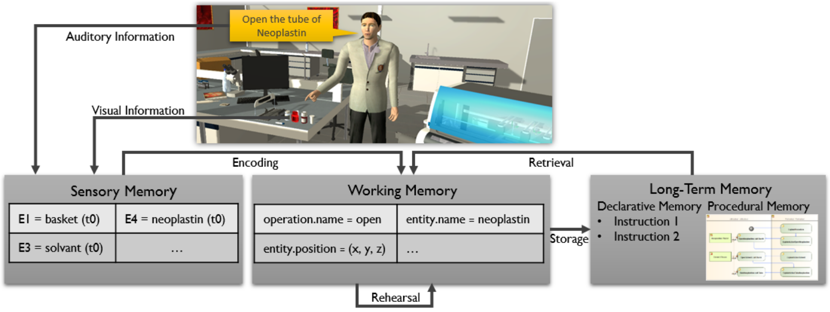

Fig. 1. Formalization of the encoding and structuring of instructions in the memory.

In this work, we therefore distinguish three structural components in human memory in which a sequence of cognitive processes is implemented to process information (encoding, storage, retrieval). The first operation involved in the information processing is the encoding of information. It is the transformation of incoming stimuli from the virtual environment and the virtual tutor to a formal representation that can be stored in the working memory. As mentioned previously, incoming stimuli are visual (set of objects in the student's field of view) and auditory (uttered by the tutor). Only prominent information (e.g. objects that have been highlighted by the tutor) is transferred from the sensory memory to the working memory. The working memory stores and manipulates information based on the content of the sensory memory and the long-term memory (prior knowledge). The level of complexity of stored information in the working memory depends on the student's prior knowledge (by complexity of information we mean the level of the formal representation in MASCARET hierarchical formalism). This prior knowledge is retrieved from the long-term memory. The transfer of some knowledge from the working memory to the long-term memory, takes place when the student completes an action [8].

This student model is used as an input in the tutor behavior.

3.2 Tutor Behavior

The tutor behavior takes into account the actions done by the learner and the inferred student model to adapt the execution of the pedagogical scenario. This adaptation can be a modification of the student model (modification of the memory content) and/or the execution of a pedagogical action. The decision making of the tutor behavior is represented in Figure 2.

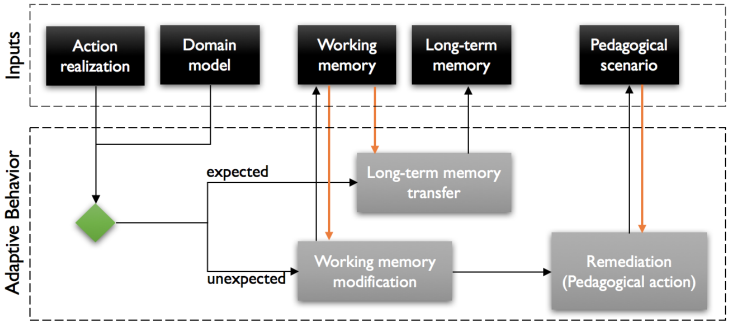

Fig. 2. Tutor behavior decision making.

The execution of a pedagogical scenario is a set of interaction between the tutor and the learner. As explained in section 2, the tutor actions (pedagogical actions) are realized through the interface, and this latter is also able to recognize the actions realized by the learner in the context of this interaction.

Our tutor behavior categorizes the actions done by the learner, based on two types of actions:

1. related to the domain model: an action can be either a domain action on a specific object or an answer to the tutor's questions. The tutor relies on the domain model to check if these actions are considered as errors or not.
2. related to the interaction: actions done by the learner can also be a feedback to the tutor's action (e.g. a facial expression, a question, observing the environment or an inaction). In this case, instead of using the domain knowledge, the tutor evaluates whether this feedback is negative or not.

If the learner's action is considered as an error or as a negative feedback, this means that this action is unexpected in the context of the executed scenario. In this case a new pedagogical action is needed and the content of the learner's memory must be reevaluated.

For example, if according to the pedagogical scenario the tutor explains the next action that the student has to do, we instantiate two chunks in the working memory, one for the `Action` and the other one for the `Entity`. If the student realizes an unexpected action (for example he/she shows a negative facial expression), then the tutor behavior considers that the student does not know the object position, contrary to what the tutor inferred. In this case the tutor remedies to this situation by re-evaluating the content of the student's working memory and then realizes a new pedagogical action to highlight the object.

4 Conclusion and Future Work

The model that we propose here allows an embodied conversational agent, playing the role of tutor, to execute a predefined pedagogical scenario written by a trainer in a virtual environment and especially to adapt its execution according to the individual evolution of students. To do this, the ECA infers the student's knowledge by estimating the content of his/her memories involved in procedural learning. The tutor behavior that we propose is a simple behavior that allows us to show the usability of the memory model that we have implemented to define a pedagogical behavior. In the same way that in MASCARET it is the trainer who describes the pedagogical scenario using a dedicated language (based on UML activities), we consider that it would be more interesting if it is the pedagogue which describes the tutor's behavior using the same language. We aim to make the concepts defined in our model accessible and formalized in this language.

In order to evaluate the impact of our model on the student's performance, we plan to carry an experiment that will involve two groups of participants. In the first group, a non-adaptive virtual tutor will be present in the virtual environment. The non-adaptive tutor will apply a single pedagogical scenario during repetitions. If the student asks for help, the tutor announces the action to be performed, its goal and highlights the object to manipulate. In the second group, an adaptive tutor will guide the learner. Based on our model, the tutor will be able to adapt the execution of the pedagogical scenario according the evolution of the learner's level of expertise. In this experiment, we expect that learners interacting with an adaptive tutor perform the procedure without errors and without the need for help, earlier than those who are interacting with a non-adaptive tutor.

References

1. Anderson, J.: A spreading activation theory of memory. Journal of verbal learning and verbal behavior **22** (1983) 261–295
2. Corbett, A., Anderson, J.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction **4** (1995) 253–278
3. Atkinson, R., Shiffrin, R.M.: Human memory: A proposed system and its control processes. In K. W. Spence and J. T. Spence (Eds.), The Psychology of learning and motivation: Advances in research and theory (vol. 2). (1968) 89–105
4. Anderson, J.: The architecture of cognition. 2nd edn. Psychology Press (2013)
5. Chevaillier, P., Trinh, T., Barange, M., Devillers, F., Soler, J., De Loor, P., Querrec, R.: Semantic modelling of virtual environments using MASCARET. In: Proceedings of the 4th Workshop SEARIS, IEEE VR, Singapore. (2001)
6. Le Corre, F., Hoareau, C., Ganier, F., Buche, C., Querrec, R.: A pedagogical scenario language for virtual environment for learning based on uml meta-model. application to blood analysis instrument. In: International Conference CSEDU. (2014) 301–308
7. Niewiadomski, R., Bevacqua, E., Mancini, M., Pelachaud, C.: Greta: an interactive expressive eca system. In: 8th International Conference AAMAS. (2009) 1399–1400
8. Ganier, F.: Factors Affecting the Processing of Procedural Instructions: Implications for Document Design. IEEE Transactions on Professional Communication **47**(1) (2004) 15–26