CENTRALIZE THE TUTORING PROCESS IN INTELLIGENT TUTORING SYSTEMS

Alke Martens (martens@informatik.uni-rostock.de)
Department of Computer Science, University of Rostock,
Albert-Einstein-Str.21, 18059 Rostock, Germany

KEYWORDS Intelligent Tutoring System, Case-based Learning, Web-based Learning

ABSTRACT
Intelligent Tutoring Systems (ITS) can look back on a long tradition. During the last years, the core architecture of ITS’ has not changed significantly. It consists of the models ‘expert knowledge’, ‘pedagogical knowledge’, ‘learner model’ and ‘user interface’ and a set of components for steering the interaction. These components differ a lot; they are domain and implementation dependent. An analysis of the functionality and role of these components in some ITS has led us to the assumption, that one centralized and formally describable tutoring process model can take the place of these components.

1 INTRODUCTION

Intelligent Tutoring Systems (ITS) can look back on a long tradition. In the early 1970s, Carbonell made the first attempts to combine Artificial Intelligence (AI) and Computer Aided Instruction (CAI). Together with Collins he developed the system SCHOLAR (Carbonell 1970) that has been used to teach South American geography in a mixed initiative dialogue. According to Lelouche, this first kind of ITS can be distinguished from the later developments. He differentiates (Lelouche 1999) between the first generation, called “Domain Expert” Intelligent Computer Aided Instruction Systems (ICAi), and the second generation, the “Educational Psychologists” ICAI. Only the second generation are what we perceive today as ITS. More than thirty years of development separate us from the first approaches towards ITS and a multitude of systems have successfully been implemented, tested, and approved in classroom settings and at universities. In the last years, a common agreement about the core models of an ITS seems to be reached in the ITS community. This classical architecture consists of at least four models, i.e. an expert knowledge model, a pedagogical knowledge model, a model that represents the learner in the system and finally a user interface model. Each of those models has become or stems from a certain research branch in computer science. Expert knowledge model and pedagogical knowledge model, both of which can be seen as a kind of ‘expert knowledge’, are part of AI research. The development of learner models has grown towards an own scientific branch in AI, as well as in psychology and cognitive sciences. User interfaces are e.g. studied in various application domains in the area of ‘human computer interfaces’ (HCI).

Static knowledge vs. dynamic interaction process
Expert knowledge, pedagogical knowledge and learner model contain static knowledge in the ITS. They are used to store information and to provide information. Compared with a scenario, where a teacher or tutor interacts with at least one learner, both expert and pedagogical knowledge model resemble the expert background of the teacher or tutor. However, textbooks
could also provide this ‘pure’ knowledge. What makes the tutor-learner interaction valuable can be described as the dynamic interaction process between both participants. In the optimal situation, the tutor acts and reacts according to the learner’s progress and needs, his level of knowledge and his performance in the actual context. Whereas in human-human interaction, the tutor has the possibility to grasp the learner’s knowledge and feeling via communication, gestures and mimic, the ITS, based on a conventional computer, can only reason about the learner’s progress via the set of information gathered during the problem solving process, the learner’s performance in checkpoints or his questions and errors.

The dynamic process of interaction between the computer program and the learner is ruled by another part of ITS. In the classical ITS architecture it takes part as exchange between the existing models (see figure 1 in the next section). Each ITS requires a set of components (at least one component) that steers and controls the exchange between the core models and between the learner and the system. Most of the time these components are treated like ‘step children’ of the ITS – they are not described in detail, they are often domain dependent, not reusable, and sometimes they are only ad hoc implementations. However, the functionalities they have to fulfill differs not significantly between the different ITS. Moreover, it reflects the really important parts of tutor-learner interaction in the computer system. Thus, we have developed one component that bundles all of these functionalities and interacts with the core models of the ITS. This component is formally described and thus can be perceived as domain independent and reusable.

2 CLASSICAL ITS MODELS

As we have mentioned in the introduction, the classical ITS models are the expert knowledge model, the pedagogical knowledge model, the learner model and the user interface model. We will now give a short insight into the role of each of these components in the ITS, except the user interface, whose role is obviously the interaction interface of the learner with the ITS. We have compared a lot of ITS, however, due to space limitations, we will only show the results of the comparison.

The expert knowledge model contains the expert knowledge of the application domain. Typically, it consists of an ontology of the application domain and of a rule base. Those are also the main constituents of an expert system, however the requirements for an ITS’ expert knowledge base are a bit different. Whereas the expert system must offer the correct conclusions given an input, the expert knowledge base of an ITS must be able to show the whole problem solving process, as e.g. Rickel notes: “…structure and relationships needed for tutoring are much greater than for typical problem-solving expert systems.” (Rickel 1989). The knowledge of an ITS expert base must e.g. allow a comparison between the learner’s solution of a training case and the expert’s optimal solution.

The pedagogical knowledge model contains the didactical strategy of the ITS. The pedagogical knowledge is used to adapt the learning material to the learner, i.e. to the information stored in the learner model and to the learner’s behavior during his work in the training case. The model’s functionalities can be summarized as ‘what to present when and how’, and sometimes also as ‘when to interrupt’. Some systems combine expert and pedagogical knowledge in one large
expert knowledge base, e.g. (Lelouche and Morin 1998). Their argument is that the pedagogical or didactical structure of the domain knowledge is a natural part of the expert knowledge. The learner model contains information about the learner. Whereas the expert knowledge model and the pedagogical knowledge model are more or less fixed regarding the amount of information stored in the databases, the learner model’s content increases according to the learner’s growing amount of knowledge, and changes due to his changing expertise during a set of sessions with the ITS. The learner model contains that part of the information the ITS has available about the learner.

The interaction between these models and between the learner and the system is performed by a set of components. These components can be grouped as: interaction components that steer the exchange of information between the expert knowledge model, the pedagogical knowledge model and the learner model; adaptation components that adapt the information displayed to the learner (in figure 1 distinguished in ‘presentation’ and ‘feedback/correction’); diagnosis components that have the responsibility to diagnose the learner’s answers and to evaluate his performance. Our comparison of ITS has shown, that this is also close to stores, flows and processes distinguished in the Learning Technology System Architecture, described e.g. in (LTSA 2002). Excluded from this analysis are pedagogical personalized agents, like e.g. ADELE realized by (Rickel and Johnson 1999). Some ITS provide an additional component which is most of the time called the exercise generator. The purpose of this component is to automatically generate exercises, given an extensive expert and pedagogical knowledge base, e.g. in the System D3 Trainer (Reinhard 1997). However, not all kinds of learning strategies lend themselves for an automatic exercise generation. For example in case-based learning, constructing a training case requires more information than an expert knowledge base can provide, because the application domains are often ill-structured and highly complex. The ITS we have developed based on the new architecture is a case-based ITS, thus we have excluded the exercise generator. Figure 1 shows a scheme of a classical ITS architecture.
3 THE TUTORING PROCESS MODEL

The functionality of the mentioned components, i.e. interaction, adaptation and diagnosis, can be performed by one single model. This model can be described as a formal model, and thus facilitates communication and exchange, bridging the gap between different application domains and different implementation languages.

Introduced into the ITS it changes the architecture towards a set of models that provide the static information, i.e. the former core models expert knowledge model, the pedagogical knowledge model, the learner model, and the tutoring process model that provides the dynamic in the system. The ‘new’ architecture is shown in the figure 2.

![Figure 2: Architecture of ITS](image)

The tutoring process model is a flexible model that steers the adaptation of the contents displayed to the learner. For this purpose it consists of adaptation mechanisms that work hand in hand with the information stored in the learner model. Additionally, it uses the expert knowledge model to adapt the training case’s development to the learner’s progress and to the logical coherence of the training case itself. The tutoring process model for generating feedback and help uses the expert knowledge in a context dependent way, taking the actual knowledge level of the learner into account. Also, it is responsible for updating the information in the learner model according to the information displayed to the learner and to his responses. A direct interaction or exchange of information between the classical components is no longer required. The tutoring process model uses the pedagogical and the expert knowledge to adapt the contents level of difficulty to the learner. This will be explained in the following subsection.

**Case-based Training**

In case-based training, the learner is confronted with a complete story, often prepared in a narrative and didactically elaborate fashion. The learner takes over a role in the training case, e.g. in a medical scenario he has to act like a physician. A training case in this context can be for
example the treatment process of a certain patient.

From the learning psychology perspective, a training case supports two cognitive processes, i.e. the process of general knowledge application and the process of diagnostic reasoning (Martens 2003). Thus, one part of the learning is to answer questions, to interact with multimedia elements, etc. Another part is to continuously adapt the diagnosis to the new amount of information gathered. And last but not least, the choice of correct subsequent steps reflects knowledge application in the training case. The tutoring process, acting as much like a human tutor as possible, uses these three kinds of information about the learner’s behavior and progress in the training case to continuously adapt new information, feedback and help to the learner’s actual amount of knowledge.

Generally, training cases can be developed in three granularities, according to the learner’s level of expertise. A guided training case provides no choice of next steps, thus it is only useful for beginners in the application domain or for demonstration of training cases in classroom scenarios. The free or unguided training case is the opposite. Here, the learner has all steps every time available. This might be appropriate for an expert, however a sub-expert or an advanced student might be swamped with the amount of steps to choose. Thus, it is reasonable to develop training cases at an intermediate and adaptable level of guidance – the half-guided training cases. Here, the amount of next steps is computed during runtime by the tutoring process model and takes into account the information the learner has gathered so far as well as the logical coherence of the training case’s story line.

According to the learner’s level of expertise in the application domain and also in working with computers, he has the possibility to choose among different representations of a training case. During his work with the case, the level of difficulty is continuously raised or lowered according to his performance. I.e. if he makes a lot of mistakes, the tutoring process provides more information and restricts the amount of available next steps. Additionally, help can be provided. If the learner’s performance is very good, e.g. he answers all questions correctly, notes the correct diagnoses and chooses the appropriate steps, the level of difficulty can be raised, i.e. a broader choice of next steps can be displayed and less detailed information given.

4 CONCLUSION

In an interview with an AI journal, John Self has criticized that ITS have a lack of formal methods and are based on a small theoretical basis (Self 1999). The new perspective on the ITS, centralizing the tutoring process model, is a step towards exchangeability of components of the ITS. It provides a theoretical basis to one of the main aspects of ITS, i.e. the dynamic process of interaction and adaptation that has to take place between a tutor and a learner, even if the tutor is only a computer program.

The tutoring process model has its background in the ITS “Docs ‘n Drugs – the virtual hospital” (Martens & Bernauer et.al. 2000) that has been developed at the University of Ulm (DOCS ‘N DRUGS 2000). The project that is part of the “Virtual University Baden-Wuerttemberg” started in 1998 with the focus of web- and case-based learning in clinical medicine. Students of the University Hospital Ulm are working with the system as part of their regular curriculum since the early 2000. Starting the development of a new ITS, we have made some investigations of existing ITS, with the aim to reuse components and to adapt parts of existing systems, if possible.
However, we found out that apart from the core models, the main interaction components of existing ITS are not reusable, not formally described, and not domain independent, even if their functionalities are quite equal. This has led us to the development of the tutoring process model, and to the change of the perspective of ITS architecture. The tutoring process model is now part of “Docs ‘n Drugs”. Currently, another prototypical ITS is developed. It shows that the tutoring process model allows to embed a strategy to reason about the learner’s mental plans in the training case and to use this knowledge to adapt the training case, feedback and help accordingly (Martens & Uhrmacher 2002).

References