Abstract. To reduce the prohibitive efforts associated to problem-solving ITS, a domain-independent framework can be developed. In this paper, we compare ASTUS to the MTT architecture with examples drawn from a scatter plot tutor. We show that both approaches are similar, but that an ASTUS tutor can exhibit a more sophisticated tutoring behavior thanks to its “pedagogical knowledge model”.

Keywords. Knowledge representation, Problem-Solving ITS frameworks.

Introduction

Problem-solving Intelligent Tutoring Systems (ITS) are generally associated with prohibitive efforts; to reduce them, a framework that offers modules and tools independent of a given task domain can be developed. A well-known example is the Model-Tracing Tutors (MTT) architecture, and more specifically the Tutorial Development Kit (TDK) that implemented the first Cognitive Tutors (CT) [2]. We designed the ASTUS framework [7] following the four-modules ITS archetype [9], aiming to go a step further in factorizing domain-independent components. To do so, we use a “pedagogical knowledge model” of the task domain designed to be exploited by a domain-independent pedagogical module. Heffernan [5] has developed a pedagogical module over the MTT architecture for the algebra task domain, but its generalization is limited by TDK’s cognitive models lack of potential for scrutiny.

Because they share fundamental features, the MTT architecture and ASTUS face a similar limitation: they can be applied only to well-defined domains. We illustrate our framework comparison with a scatter plot tutor, developed with TDK [3] and its replica built using ASTUS.

1. The Scatter Plot Task Domain

A scatter plot is a Cartesian graph that displays two-variable data. The task consists of establishing a first variable as the independent variable, then scaling it along the abscissa (x-axis). Then, a second, dependent variable is scaled along the ordinate (y-axis). Finally, the data are turned into a collection of points, the value of the independent/dependent variables giving the position on the x-axis/y-axis.

1 Corresponding Author: jean-francois.lebeau2@usherbrooke.ca
2 Throughout this paper, we use Van Lehn’s lexicon [8] to characterize such systems.
3 TDK tutorial slides http://c tat.pact.cs.cmu.edu/pubs/TDKTutorialSlides.ppt
We selected the scatter plot task domain because the complexity of its knowledge components and learning environment was appropriate for this comparison. Moreover, Ryan S.J. Baker gave us access to the TDK tutor used in the classroom, the underlying model and a sample of learners’ data\textsuperscript{4} that we transformed into simulation data.

2. Comparison Methodology

The knowledge representation approaches are compared using these criteria we synthesized from a wide coverage of the ITS literature while designing ASTUS:

- The kinds of knowledge components available to model the task domain.
- The coupling between the knowledge model and the learning environment.
- The pedagogical learning content \textsuperscript{5} that can be attached to the model.
- The knowledge model’s potential for scrutiny.
- The tools offered to reduce the efforts required to model a task domain.
- The capacity to build pedagogical units from knowledge components.

The tutoring behaviors are compared with criteria drawn from VanLehn’s \textsuperscript{8} synthesis of the behavior of ITS:

- Minimal step feedback.
- Error-specific feedback.
- Next step help.
- Next task selection.

3. Comparison Results

3.1 Comparison of the Knowledge Representation Approaches

Both approaches are centered on a procedural knowledge component that is the object of tutoring: production rules in TDK and complex procedures in ASTUS. A first distinction is that the rules model a cognitively plausible explanation for a given step, whereas procedures reify a pedagogically relevant path from a goal to the steps needed to satisfy it. A second divergence is that ASTUS uses many kinds of knowledge components (including chunks, procedures, goals, queries and rules) where TDK uses only two: production rules and Working Memory Elements (WME). Thus, ASTUS’ procedures are composed of “first-class” components that can be inspected by a domain-independent module. In opposition, even if a TDK model as a whole is a “white box”, each production rules is a small “black box”. ASTUS’ models also contain “black boxes” to formalize basic domain-specific inferences that are required to have an operational model. These inferences are not tutored, but may be pointed to as pre-conditions of a step. A third difference is the coupling approach between the KB and the learning environment. Instead of having WMEs tied to UI elements, we propose a MVC-based approach in which the KB acts as the Model \textsuperscript{4}. This approach is more flexible when modeling steps than TDK’ SAI events and facilitates rich forms of feedback. Both approaches: support simple pedagogical learning content, such as message templates associated with knowledge components, offer a limited developing environment and can group knowledge components in pedagogical units.

\textsuperscript{4} Encoded using PSLC’s DataShop format (https://pslcdatashop.web.cmu.edu/)
3.2 Comparison of the Tutoring Behaviors

As CT’s tutoring behavior has proven successful in the classroom [6], a method to show the benefits of our framework is to support it with less efforts required for an author. We implemented a pedagogical module that includes: flag feedback, error-specific messages, and next-step hint messages. ASTUS is able to generate some of these messages by scrutinizing the model in function of the current situation. The ASTUS scatter plot tutor did not require invalid procedures for various errors modeled with buggy rules and their associated messages. ASTUS can generate messages for error such as forgetting to enter a scale value or to add a data row’s point and inverting steps required to calculate a variable’s scale. Also, using the learner’s guidance and its basic learner module, ASTUS can vary the level on which help is given, navigating its hierarchical procedural knowledge components. For example, in the scatter plot tutor, a help message can hint to the next scaling step or to the overall process of scaling a variable. Moreover, ASTUS can enhance the basic forms of feedback with UI effects like performing a step’s demonstration [4]. Both CT and ASTUS can automate mastered steps using their learner model module and offer a macro-adaptation task selection strategy which relies on skills defined with knowledge components.

Conclusion

We have compared ASTUS to the MTT architecture, to show that they are similar, but that our scatter plot tutor can exhibit a more sophisticated tutoring behavior using its “pedagogical knowledge model”, a model which capacity to be examined is designed to support pedagogical observations. Our next steps are to develop our domain independent modules and to make ASTUS available in a way similar to CTAT [1].

References