An Authoring System for Instructionally Designed Tutoring Processes

Alexander Seitz*, Matthias Dannenberg*, Hubert Liebhart§
*Dept. of Artificial Intelligence, University of Ulm, 89069 Ulm, Germany,
{seitz,dannenberg}@informatik.uni-ulm.de
§Faculty of Medicine, University of Ulm, Albert-Einstein-Allee 7, 89081 Ulm,
hubert.liebhardt@medizin.uni-ulm.de

Abstract
This paper describes the design and implementation of a
domain independent authoring tool for tutoring systems.
The presented work incorporates standard authoring
features, including a hierarchical development of learning
material, but also supports users in a didactically suitable
design of the tutoring process. Requirements for the
realization of Goal Based Scenarios are fulfilled by a
number of implemented add-ons.

Introduction
E-Learning plays a significant role in the field of
education and training today. In this connection, the
success and acceptance of tutoring systems depends on the
quality of the mediated knowledge and skills as well as
the quality of how these things are mediated. By now, a
considerable assortment of authoring systems exists [1],
which support authors well in developing high quality
tutoring material. Nevertheless authors still ask for special
help for a suitable instructional design of their work. This
applies particularly in the field of problem-oriented
learning. We designed and implemented an authoring tool
to fill this important gap.

The theoretical basis of our system is the instructional
design theory of Goal Based Scenarios developed by
Roger Schank [4]. Referring to this approach, the
authoring tool guides case engineers in designing cases to
teach procedural knowledge to solve problems on a high
level. In a first step the case engineer identifies the target
skills and defines concrete learning objectives, which are
realized by defining an exact mission and its submissions.
The mission is embedded in a realistic, authentic and
enriched cover story that supports the learner to associate
him with the real case situation. Finally, two main focuses
of the cognitive learning process are supported, namely
the discovering of information and the explanation of
theoretical and procedural knowledge. Concretely, tasks
and learning materials are realized by various learning
operations. For example, in the medical domain of Docs
’n Drugs [2], those operations include searching
information according to diagnostic reasoning, answering
questions for relevant examinations and choosing
alternatives to find the correct diagnosis.

Usually, a tutoring case can be used for different
target groups with special learner profiles. Nevertheless, it has to be adapted to the special
requirements of each group. Thus we designed our
authoring tool to support structuring a tutoring case
for allowing the selection of specific case-scenarios
and difficulty levels.

Basic authoring features
The basis of a tool for authoring instructionally
designed tutoring systems is a set of basic features
that allow building appropriate special support
modules upon them. We implement a hierarchical
system of modules and frames. Frames form the
leaves of the hierarchy and contain presentation and
interaction elements shown to the learner. See Figure
1 for an example. Contents can be fed into single
frame slots on a declarative level, while the author
separately specifies the presentation and interaction
elements that display these contents. Concrete
tutoring system implementations are generated based
on these specifications, including web-based systems and standalone applications. By this procedure, contents become resistant to individual demands and future technological developments of tutoring systems. The nodes of the hierarchy are modules, consisting of frames or other modules that pertain to each other in content. For each module, common properties of its elements can be defined in attribute editors. In order to specify which other frames or modules are reachable from a given frame or module, the author is able to draw arrows between them. These arrows can be enriched by attribute-based conditions, specifying when the associated navigation possibility becomes active. For example, in Docs 'n Drugs the module for a first acquisition of patient data is typically connected to a module where the medical student has to decide which technical examinations have to be done necessarily. The student is not able to reach the second module until he has performed a specified set of anamnesis and body examinations. In a similar manner, an author can create conditional branches within a navigation possibility. For that he draws arrows from a condition node to the possible alternatives for the navigation. A special condition editor feeds the conditions into the system.

**Instructional design components**

Before an author builds concrete frames of a tutoring case, the learning objectives of that case must be identified. We offer possibilities to formulate learning objectives and assign them to the tutoring case, to modules, or finally to individual frames. Based on the assignments made, the system can build a learning objectives map that helps the author to get an overall view of the didactical structure of his tutoring case.

Learning objectives are realized by a framework of missions and submissions that a learner has to perform. For that, our system allows the definition of instructional design patterns, which form the basis of such frameworks. Instructional design patterns are implemented by configurations of named modules and basic navigation structures, where each module can be supplemented with frames that realize common case aspects. For example, an instructional design pattern that is typical for Docs 'n Drugs includes a module for the standard acquisition of patient data plus frames for a first differential diagnosis, a module for performing specific examinations and making the final diagnosis, and a module for finding the appropriate therapy. Those patterns can be stored, integrated into the user interface of the authoring system, and reused for the design of other tutoring cases.

We support the realization of two important focuses of the cognitive learning process: the discovering of information and the explanation of theoretical and procedural knowledge. For that, frames can be associated with facts like for example patient data. Furthermore, a general rule editor allows relating facts with each other or with procedural knowledge. Thus, patient data can be related to diagnoses, while diagnoses together with patient circumstances can be related to appropriate therapies. Based on that, tutoring systems can realize facilities to discover and explain the formulated facts and relations. For example, Docs 'n Drugs does this by displaying found relevant patient data in a finding map and asking for the relations in special differential diagnosis and therapy tools.

In order to prepare tutoring cases for the use by different target groups, it has to be kept modular and variable adequately. The use of modules within instructional design patterns and their association with general learning objectives enables a specific selection of those modules by the tutoring system for a given target learner group. Additionally, our system allows formulating different variants of a single frame, frame element, or conditional branch, each of which can be associated with facts like user levels or learning objectives. Those associations together with user preferences or user modeling components allow an individualized configuration of the tutoring case.

**Conclusion**

The system is implemented in Java and supports jar plugins to extend and adapt its functionality to author needs. An application generator allows different environment realizations like client-server, CD, or web-based products. An XML interface allows the exchange of tutoring data between applications. As far as medical tutoring cases are concerned, we are also involved in standardization efforts [3] for that kind of material.

Ongoing work will focus on the further development of a library of relevant instructional design patterns. This will happen together in corporation with our internal and external application partners.

**References**


