ONTOLOGY BASED FEEDBACK GENERATION IN DESIGN-ORIENTED E-LEARNING SYSTEMS

Harrie Passier and Johan Jeuring
Faculty of Computer Science, Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
harrie.passier@ou.nl/ johan.jeuring@ou.nl

ABSTRACT

One of the essential elements needed for effective learning is feedback. Feedback can be given to learners during learning but also to authors during course development. In the current generation of eLearning systems automatically produced feedback is sparse, mostly hard coded, not very valuable and almost only used in question-answer situation. Valuable feedback, for example produced by a human tutor via e-mail, is often possible but this introduces delays and is time consuming. We want to develop mechanisms, based on ontologies, to create a rich supply of feedback, not only in question-answer situations but also in the context of design oriented education. With ontologies we specify (1) the knowledge to be learned (domain and task knowledge) and (2) how the knowledge should be learned (education). We will develop algorithms with which we automatically create valuable feedback to learners during learning, and to authors during course development. Furthermore, we will develop a formalism with which course developers can specify domain and task specific feedback to learners. In this paper we present our research plan.

KEYWORDS

Electronic learning environments, feedback, ontology, algorithms, design oriented education

1. INTRODUCTION

Feedback is used in many learning paradigms. The concept of feedback is crucial in educational psychology. It is an accepted psychological principle that one of the essential elements needed for effective learning is feedback. Knowledge of results is required to assess progress, correct errors and improve performance (Mory, 2003). Feedback describes any communication or procedure given to inform a learner of the accuracy of a response, usually to an instructional question. More broadly, feedback allows the comparison of actual performance with some set standard of performance. In technology-assisted instruction, it is information presented to the learner after any input with the purpose of shaping the perceptions of the learner. Information presented via feedback in instruction might include not only answer correctness, but other information such as precision, timeliness, learning guidance, motivational messages, background material, sequence advisement, critical comparisons, and learning focus (Mory, 2003).

In a classroom learners and teachers can easily interact, i.e. students can freely ask questions and teachers usually know whether their students understand (basic) concepts or problem solving techniques. Feedback is an important component of this interaction. Furthermore, educational material can be continually improved using information from the interaction between the lecture and the learners, which results in a more efficient and effective way of course development.

There is a frequent lack of feedback in electronic learning environment (or eLearning) courses in higher education (Mory, 2003). Almost all feedback is related to question-answer situations and is hard coded. Exceptions are the environments based on social constructivism (Duffy, 1996). In such environments learners solve complex problems through social negotiations between equal (human) peers in a contextual setting. Feedback occurs in the form of discussions among learners and through comparisons of internally structured knowledge (Mory, 2003). This type of feedback introduces delays and is time consuming. In specialized applications, for example programming and design editors, much more feedback is given. Some of these (for example JBuilder enterprise version 8) especially address feedback on the syntactic level. Semantically rich feedback can only be found in specialized design environments. An example can be found in (Fischer, 1993) in which an environment for kitchen design is described. Unclear is how much effort is needed to realize such a system and how general the (interference) mechanisms are, i.e. what has to be done if the system is used in another domain. Other examples of feedback in eLearning systems can be found in (Hummel, 2001) and (Martens, 1998)

Many eLearning systems are based on ontologies. Aroyo et al (Aroyo, 2002 (1) (2)) describe an authoring tool based on ontologies to (1) support the development of domain and task ontologies and (2) support and perform (semi) automatic courseware authoring activities. Feedback is given in the form of hints and recommendations. Jin et al (Jin, 1999) describe an authoring system that uses ontologies to produce feedback (error, warning and suggestion) for an author. Both a domain ontology as well as a task ontology are used. The ontologies are enriched with axioms, and on the basis of the axioms messages of various kinds can be generated when authors violate certain specified constraints. Literature about giving semantically rich feedback to learners is sparse and is only found in specialized design environments. See for example (Fischer, 1993).

In our research we want to develop generic, domain and task independent, feedback mechanisms that produce semantically rich feedback to learners and authors during learning and authoring. We distinguish three types of feedback: (1) feedback given to a student during learning, which we call *student feedback, (2)* feedback given to an author during course authoring, which we call *author feedback* and (3) feedback from a group of learners who study a course to an author, which we call *group feedback*. With group feedback an author may be able to optimize his/her course. We will develop generic feedback mechanisms where ontologies are arguments of the feedback engine. This is important, because the development of feedback mechanisms is time consuming and specialist work, and can be reused for different ontologies. Besides generic feedback mechanisms we will also develop mechanisms by means of which authors can define domain and/or task specific feedback. We will focus our research on design environments for Computer Science courses, especially design environments in which artefacts can be made using languages like Unified Modeling Language (UML) and Object Constraint Language (OCL) (Warmerdam, 1999).

In this paper we introduce our ideas about an eLearning system, in the field of design-oriented education, that produces semantically rich feedback to authors as well as to learners. In Section 2 we explain our ideas with two examples and we give a sketch of the functionality and architecture of the system we imagine. The examples are related to the domain communication technology. In Section 3 we describe our research questions. Finally in Section 4 we draw our conclusions.

2. A DESIGN ENVIRONMENT THAT PRODUCES SEMANTICALLY RICH FEEDBACK

We imagine an eLearning environment for computer science courses, in which: (1) learners are able to design artefacts of certain domains using different types of languages, and (2) authors are able to develop courses. Learners as well as authors receive semantically rich feedback during learning, designing artefacts and developing courses based on different ontologies, for example a domain, a task, an educational and a feedback ontology. First we give two examples to explain our ideas: one about a learner who develops artefacts in a player, and one about an author who develops course material in an authoring tool. After that we give a short description of the functional architecture.

2.1 Example of a player

A student first has to learn the concept (communication) network. Assume that a network consists of links, nodes, a protocol and a protocol driver. Each of these concepts consists of sub-concepts. The domain ontology 'communication technology' represents these in terms of a vocabulary of concepts and a description of the relations between the concepts. On the basis of an education ontology, which describes the learning tasks, the student is asked to list the concepts and relate the concepts to each other. Feedback is given about the completeness and correctness of the list of concept and relations using different dialog patterns.

In a second step the learner is ask to design a part of a local area network (LAN) using the network model developed during the first step. Instead of concepts, concrete instantiations must be chosen and related to each other. The learner gets feedback about the correctness of the instantiations and the relations between the concepts. Some protocols for example need a specific network topology. There are various sequences of activities to develop a network, each of them with its own particular efficiency. The student gets feedback about the chosen sequence of activities on the basis of the task ontology. Further, the student receives different types of feedback, for example corrective/preventive feedback, critics and guiding. All these feedback types are further customized to the learning style of the learner.

2.2 Example of an authoring tool

An author develops and optimizes a course. He/she has to choose, develop and/or adapt particular ontologies and develop related material like examples, definitions, etc. Based on analyses of the domain, education and feedback ontologies, the author gets feedback, for example about:

- Completeness: A concept can be used but not defined. Ideally, every concept is introduced somewhere in the course, unless stated otherwise already at the start of the course. This error can also occur in the ontology for the course.
- Timeliness: A concept can be used before its definition. This might not be an error if the author uses a top-down approach rather than a bottom-up approach to teaching, but issuing a warning is probably helpful. Furthermore, if there is a large distance (measured for example in number of pages, characters, or concepts) between the use of a concept and its definition in the top-down approach, this is probably an error.
- Synonyms: Concepts with different names may have exactly the same definition.
- Homonyms: A concept may have multiple, different definitions.

The author defines specific feedback for the composed concept network, because group feedback shows that this composed concept is experienced as difficult: extra support in the form of feedback is needed. When the author changes the domain ontology, the generic feedback mechanism remains working.

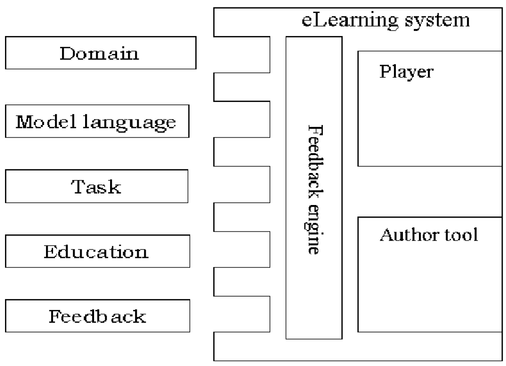
2.3 Functional architecture

To produce semantically rich feedback the system should contain several types of knowledge. To represent this knowledge we make use of ontologies At this moment, we distinguish knowledge about:

- **Domain** – For example: Communication Technology, or Distributed Programming
- **Modelling language** – For example: UML/OCL
- **Task/method** – For example: the sequence of phases during design in which an artefact is built
- **Education** – For example: concept learning, problem solving, examples and definitions.
- **Feedback** – For example: different types of feedback and patterns/phases during dialogs.

Figure 1 gives the architecture of an eLearning system that supports a generic feedback mechanism

Figure 1. Functional architecture



The eLearning environment consists of three main components: a player for the learner, an authoring tool, and a feedback engine, and takes a set of ontologies as argument. The player consists of a design and learning environment in which a learner can learn concepts, construct artefacts and solve problems. The authoring tool consists of an authoring environment where the author develops and maintains courses and course related materials like ontologies, examples and feedback patterns. The feedback engine automatically produces feedback to learners as well as to authors.

The feedback engine produces generic feedback and specific feedback. Generic feedback is independent of the ontologies used and is applicable to all design activities and artefacts. Specific feedback is defined by the author and can be course, domain, modelling language or task specific. To construct feedback, the feedback engine uses the five argument ontologies. Since the ontologies are arguments, the feedback engine doesn't have to be changed if an ontology is changed for another.

The feedback engine can produce the three types of feedback mentioned. To produce student and author feedback, student and author activities are observed and matched against the ontologies mentioned. To produce group feedback information of a number of students working on a particular course is given to the author of the course. Using this information, an author may be able to optimize his/her course.

3. RESEARCH QUESTIONS

Two main research questions in this project are: (1) How can we construct an eLearning system that produces semantically rich feedback for a learner during design tasks and for an author when authoring course material, where feedback is based on a combination of ontologies and these ontologies can be changed, reused, adapted and/or extended? (2) How can we specify domain specific feedback?

There are many sub questions:

- Is a separation in distinct ontologies for domain knowledge, task/method knowledge, education and feedback knowledge meaningful and does a combination of these ontologies deliver valuable feedback to a learner in the context of design-oriented education in computer science?
- Which languages and structuring mechanisms are useful to represent knowledge types (ontologies)?
- How can we read and semantically interpret the activities of a (group of) learner(s) in the design environment?
- Is it possible to generate algorithmic feedback, for example based on grammar analysis techniques or are AI techniques needed?
- How can we specify in an efficient way generic and specific feedback during the authoring phase?
- Which classification of feedback types is valuable for the development of automatically generated feedback in design oriented eLearning systems?

Finally, we want to realize such an eLearning system as a prototype, and use this prototype for one or two courses, for example for the domain ontology 'Communication Technology' and modelling languages UML/OCL.

The focus in this research will be on the representation of ontologies using languages/grammars, grammar analysis techniques, algorithms and AI techniques to create feedback.

4. CONCLUSION

Feedback is crucial in education: it is an essential element needed for effective learning. Semantically rich feedback is sparse in most eLearning systems. In this paper we present our ideas about an eLearning system that produces semantically rich feedback for learners as well as for authors. The system we imagine consists of a generic feedback engine: different ontologies can be plugged in, i.e. they are the arguments of the feedback engine. This is important because mechanisms for automatically generating feedback are involved, and can and should be reused for different ontologies. The system supports the generation of generic as well as domain specific feedback.

An eLearning system that produces semantically rich feedback is very desirable, because feedback is crucial in effective learning, feedback is sparse in most eLearning systems, and the number of eLearning systems and eCourses is growing rapidly.

REFERENCES

Aroyo, L., et al , 2002 (1). A layered approach towards domain authoring support, *International Conference on Artificial Intelligence (ICAI'02)*, Las Vegas, USA.

Aroyo, L., et al, 2002 (2). Courseware authoring tasks ontology, *International Conference on Computers in Education (ICCE'02)*, Auckland, New Zealand.

Duffy, T., Cunningham, D., 1996. *Constructivism: Implications for the design and delivery of instruction*, in Handbook of research for educational communications and technology, D.H. Jonassen (Ed.), MacMillian Library Reference New York, USA.

Fischer, G., et al, 1993. Embedding critics in design environments, *The knowledge engineering review*, Vol 8, No. 4.

Hummel, G., Nadolski, R., 2002. Cueing for schema construction: Designing problem-solving multimedia practicals, *Contemporary educational Psychology,* Vol. 27, pp. 229 – 249.

L. Jin, W. Chen, et al, 1999. An ontology-aware authoring tool, In *Artificial intelligence in Education*, S. Lajoie and M. Vivet (Eds), IOS Press, Amsterdam, The Netherlands

Martens, R., 1998. *The use and effects of embedded support devices in dependent learning*, Phd thesis, Lemma, Utrecht, Netherlands.

Mory E., 2003. *Feedback research revisited*, in Handbook of research for educational communications and technology, D.H. Jonassen (Ed.), MacMillian Library Reference, New York, USA.

Warmerdam, J, Kleppe, A, 1999. *Praktisch UML*, Addison Wesley, Amsterdam, The Netherlands.

Johan Jeuring
Faculty of Computer Science, Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Institute of Information and Computer Sciences, University of Utrecht
Padualaan 14,3584 CH Utrecht, The Netherlands
johan.jeuring@ou.nl