

Ontology based feedback generation in design-oriented eLearning systems

Harrie Passier
Faculty of Computer Science
Open Universiteit Nederland
harrie.passier@ou.nl

25 May 2004, version 4

Content

| | |
|---|----|
| Management summary..... | 3 |
| 1 Introduction | 9 |
| 1.1 A scenario: Feedback in 2015 | 9 |
| 1.2 Aspects in the scenario | 10 |
| 2 Problem description..... | 11 |
| 2.1 ELearning | 11 |
| 2.2 Design-oriented education in computer science | 12 |
| 2.3 Feedback | 12 |
| 3 Proposed research | 14 |
| 3.1 Research questions | 16 |
| 3.2 Related work | 17 |
| 3.3 Type of research | 18 |
| 3.4 Domain and context of this research | 19 |
| 3.5 Products and schedule | 19 |
| 4 Literature..... | 20 |
| 5 Appendices..... | 22 |

Management summary

Submitted and accepted to the IADIS e-Society 2004 Conference as 'Ontology based feedback generation in design oriented e-Learning systems' by Harrie Passier and Johan Jeuring.

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 (i.e. software based) 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 presented 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 general, feedback allows the comparison of actual performance with some standard set 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, is time consuming and can't always take place. 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 will 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 asked 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 develops 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, sometimes valid depending on the context.

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 remains working.

2.3 Functional architecture

To produce semantically rich feedback the system should contain several types of knowledge. 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/didactics – For example: concept learning, problem solving, examples and definitions.

Feedback – For example: different types of feedback and patterns/phases during dialogs.

To represent this knowledge we make use of ontologies. An ontology is (Russell, 1995): (1) a vocabulary, i.e. an informal description of the concepts, predicates, functions and constants of the domain of interest, and (2) an encoding of general knowledge about a domain in terms of axioms. With axioms we describe the domain precisely and we are able to construct inference procedures to automatically derive consequences, in our case to produce feedback.

Figure 1 gives an overview of the central idea of a generic feedback mechanism:

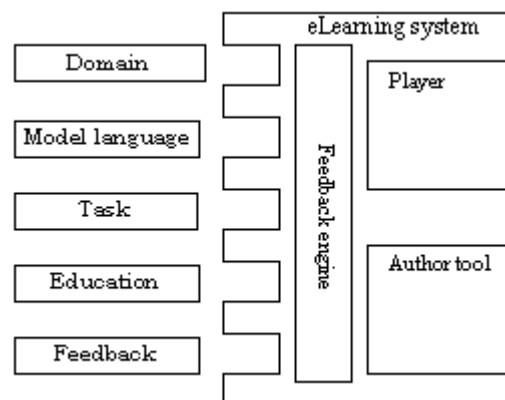


Figure 1. Functional architecture

The eLearning environment consists of four main components: a player for the learner, an authoring tool, a feedback engine and a set of ontologies as pluggable components. 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 (student, author and group feedback). 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?
- Which aspects are domain specific and which dependencies exist between the different ontologies?

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.
- Russell, S, Norvig, P., *Artificial intelligence: A modern approach*, Prentice Hall International editions, Londen,UK.
- Warmerdam, J, Kleppe, A, 1999. *Praktisch UML*, Addison Wesley, Amsterdam

1. INTRODUCTION

In this project we want to investigate the development, use and analysis of ontologies in the construction of feedback in eLearning systems. Feedback can be given to authors during course development and to learners during learning. 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 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.

Ontologies are formal descriptions of shared knowledge in a domain. With ontologies we are able to specify (1) the knowledge to be learned (domain and task knowledge) and (2) how the knowledge should be learned (education). In combining instances of these two types of ontologies, we hope that we (1) are able to create (semi-) automatically valuable generic feedback to learners during learning and to authors during course development, and (2) are able to provide the authors with mechanisms to (easily) define domain and task specific feedback to learners.

1.1. A scenario: Feedback in 2015

Peter is a bachelor student at the faculty of Computer Science of the Open University of the Netherlands. He is studying the integrating course 'Modelling and Communication Technology', in which the modelling languages UML/OCL (Unified Modelling Language/Object Constraint Language¹) and CC (Calculate with Concepts²) are used to model parts of communication technology. Now he is busy with part I, in which his knowledge about the concepts vibrations, waves, sinus function, amplitude and frequency modulation is tested. Definitions are given in the textbook. He actively models these concepts in a more formal way, he experiences his lack of *precise* knowledge about these concepts; something he doesn't discover during the self-test in which (only) definitions in a more informal way are asked.

After Peter has constructed the essential concepts in the eLearning tool, the tool directly gives *feedback* in the form of stimulating question and hints: "You declare the concept wave. By means of which other concepts is the concept wave defined?". Peter understands the hint and supplies the deficiency. During the second phase of the exercise he relates the concepts to each other. He makes use of some standard relation types and places them between the concepts. The system reacts again with feedback in the form of questions. "You have defined the concept sinus function with two parameters, frequency and phase. How does this relate with the relation make-use-of between sinus function and amplitude modulation?". Peter doesn't understand this and there starts a *dialog* between Peter and the eLearning system. After some time Peter asks for more *support*. The system *guides* Peter to the corresponding theory.

¹ See [26]

² A formal and rigorous design approach based on relational algebra [7][8]

After Peter has finished part I of the course Modelling and Communication Technology, he starts with part II in which he has to construct a UML/OCL model of a particular communication protocol. Three models of a particular communication process must be build: (1) a class model, (2) a sequence diagram and (3) a state diagram to model some dynamic aspects of the communication process. The exercise contains some new aspects in the field of communication protocols and *several solutions* are possible. Besides *feedback about correct syntactic use* of UML/OCL, he also receives *feedback on a semantical level* and the *order in which he constructs the different models*. Examples of semantical feedback are messages about errors in the OCL-model in relation to rules that are generally valid in the field of protocols, clues that indicate one or more entities are missing, and critics about the order in which the models are build.

Frans is one of the authors of the course Modelling and Communication Technology. The course is implemented in an eLearning system. Four types of ontologies form the backbone of the course. Firstly, there is an ontology in which parts of the *domain* communication technology are described. This corresponds to the theory students have to learn and the practical problems learners have to solve. The theory to be learned is connected to the domain ontology. Secondly, there are ontologies that describe the *modelling languages* UML/OCL and CC, and the related *sequences of activities*. Thirdly, there is an ontology that describes the educational patterns of concept learning and problem solving. Fourthly, there is an ontology that describes the types of feedback the eLearning system is able to deliver. Today one task must be extended and one task improved. The extension is a result of developments in the domain Communication Technology. A new type of LAN access protocol has been developed. Frans extends the domain ontology and connects the newly written theory to the ontology. The feedback to the students doesn't have to be extended, because this is generated automatically on the basis of the four ontology types and the general interference mechanisms. During the extension of the domain ontology Frans is receiving feedback about completeness (concepts that are used but not defined), timeliness (concepts are used before they are defined) and correctness (with respect to the domain ontology) of concepts. The improvement is a result of the outcome of the audit module. Until now ten learners have done the course and it becomes clear that the feedback as part of the problem-solving task should be changed. Eight learners needs more sub steps to be able to make the three models completely. Frans adds two activities to the activity sequence ontology. Also here, the feedback will be generated automatically.

1.2. Aspects in the scenario

- Ontologies are used to implement knowledge about (1) the domain of interest (in this case Communication Technology), (2) the modelling languages used (in this case UML/OCL and CC) to describe domain concepts, solve problems and valuable sequence of activities in which models can be build, (3) educational patterns (in this case concept learning and problem solving) and (4) types of feedback (for example preventive and corrective) that can be generated and different dialogs that can be happen.
- A design environment in which learners actively learn concepts and solve problems of a particular domain using formal modelling languages. The use of formal languages results in a precise definition of concepts and models, and a more rigorously understanding of these.
- Based on (the combination of) the four ontologies and learners input, learners receive semantically rich feedback to the (temporary)

modelling results in the editor during concept learning and problem solving.

- Besides one-way communication, feedback is also part of dialogs between learners and the eLearning system: feedback is adapted in an iterative way depending on learners input.
- Learners are guided through the educational material when they search for lacking knowledge.
- Besides the automatically produced general feedback an author can easily add domain specific feedback and/or task specific feedback.
- When adapting and/or extending ontologies, an author receives feedback, for example about correctness, completeness and timeliness of concepts defined.
- The general feedback mechanisms continue to work if one of the ontologies is changed. The ontologies are arguments of the feedback mechanism, i.e. the feedback mechanism is generic with respect to the ontologies.
- The author receives feedback from a group of learners who study a course, i.e. (1) information from the learners how they experience the course's content, environment and didactics, and (2) information about where the learners experience difficulties.
- There are many types of feedback. For example feedback can refer to syntax and semantic issues, can be corrective and/or preventive, and can be presented as cues or critics.

2. PROBLEM DESCRIPTION

2.1. ELearning

Education is typically interactive: a student reads, learns concepts, tries to solve problems, gets feedback on his or her solutions, and uses the feedback to make progress in his or her learning. Learning differs from individual to individual; something that is difficult for one student might be easy for another.

Online learning (or eLearning) offers new possibilities in learning: a student can get immediate feedback on solutions to problems, learning paths can be individualized, etc. Online learning is a growing business: the number of organizations working on online learning and the number of courses available on the Internet is growing rapidly: see for example the websites of the ACM and IEEE.

There exist several eLearning tools, with varying functionality and purpose. In this research we will focus on design environments (see for example [11]), especially design environments in which artefacts can be made using languages like UML/OCL and CC. These languages play an important role in the education of computer science. There exist several authoring tools for the development of electronic courses (see for example [2], [3] and [18]). In many of these tools ontologies are used to structure domain and task knowledge. One of the problems in the field of authoring systems is that building (intelligent) eLearning systems is still difficult and expensive [18].

2.2. Design-oriented education in computer science

Constructing designs plays a central role in design education of computer science. Students have to construct several types of artefacts in several domains. Examples of courses at the OUNL are: Basics of modelling, Business processes enterprise modelling, Development of information systems, Visual programming and object oriented programming with Java, and Topics in communication technology. Different domains have to be modelled (like business organization and processes, information systems and communication technology) and different languages are to be used (like DFD, PSD, NIAM, Entity-Relation modelling and UML/OCL).

Before students are able to solve problems and, as part of problem solving, to develop different artefacts, they have to acquire knowledge about: (1) the domain to model and (2) the modelling languages, methods and methodologies. Well-known learning strategies to acquire these types of knowledge are *concept learning* and *problem solving*. Concept learning is also a basic component of forming complex skills like problem solving. For an overview of both types of learning see [17] (or appendix A).

2.3. Feedback

In this paragraph we shortly explain the term feedback: first from system theory and secondly we examine feedback in the context of education and eLearning.

System theory There exists a lot of literature about feedback and control systems. Feedback can be defined as ‘that property of a closed-loop system which permits the output to be compared with the input to the system so that the appropriate control action may be formed as some function of the output and input’ [7]. For descriptions and definitions about terms like system, loop, output, input, ..., etc., see for example [7] and [25]. For more background material about feedback from system theory see appendix B.

Feedback in education Feedback is incorporated 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 to take place is feedback. Knowledge of results is required to assess progress, correct errors and improve performance [16]. In the purely instructional sense, feedback can be said to describe 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, lesson, sequence advisement, critical comparisons, and learning focus [17]. Another definition is ‘feedback is information, about behaviour or about its impact that is fed back to an individual or a group, with the intention of influencing future performance’ [23].

In a classroom learners and teachers can easily interact, i.e. students can freely ask questions and teachers know whether their students understand (basic) concepts and 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.

Feedback in eLearning There is a frequent lack of (computationally produced) feedback in eLearning courses in higher education [17]. Almost all feedback is related to question-answer situations and is hard coded. 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 [11], in which an environment for kitchen design is described. Unclear is how much effort is needed to realise 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.

An important reason for the lack of feedback in eLearning systems is that the development of such (intelligent) systems is mostly a time-consuming and specialist work [18].

There exist feedback rich eLearning environments, for example the on social constructivism [10] based environments. In such environments learners solve complex problems through social negotiations between equal (human) peers in a contextual setting. Feedback occurs then in the form of discussions among learners and through comparisons of internally structured knowledge [17].

Functions and effects of feedback Mory [17] lists as functions of feedback: (1) to motivate for increasing response rate and/or accuracy, (2) to provide a reinforcing message that would automatically connect responses to prior stimuli – the focus being on correct responses and (3) to provide information that learners could use to validate or change a previous response – the focus falling on error responses. Martens [16] mentions also as function of feedback: ‘to offer information about the learning process and the mastery of learning goals’, which is of particular interest in distance education where there is no direct contact between learners and teachers.

The effect of feedback may have two broad purposes [23]: to affect the quantity and the quality of performance. Martens mentions as possible effects of feedback [16]: (1) effects (on the sequence) of the subjects studied, (2) effects on the way students study and (3) effects on the effort and/or study time.

Types of feedback The number of feedback modes is almost infinite [16]. We give here some results of a first literature study. Mory [17] mentions the distinction between formative feedback, which modifies a student’s thinking or behaviour for the purpose of learning, and summative feedback, which assesses how well a student accomplishes a task or achieves a result for the purpose of grading. Feedback could be preventive or corrective. Tosti (e.a.) [23] makes distinction between ‘what’ (the performances itself), ‘where’ (public or private) and ‘when’ (immediately or delayed). Further, feedback can be positive or negative. Martens mentions also internal (self regulated) and external feedback [16].

Different names have been used in the literature for feedback. Examples are:

- **Feedback** – As discussed by Mory [17], Martens [16] and Tosti [23].
- **Critics** – Used by Fischer (e.a.) [11]. A critic is defined as ‘a dialog in which the interjection of a reasoned opinion about a product or action triggers further reflection on or changes to the artefact being designed’.
- **Cueing** – As discussed by Hummel (ea.) [13] in which adequate task-valid cueing is considered to facilitate both recall and interpretation of available schemata (task performance) and the construction of new schemata (learning).

We distinguish three types of feedback. Firstly, feedback is given to a student during learning. We call this first type of feedback: *student feedback*. Secondly, feedback is given to the author during authoring a course, for example during ontology development and specification of examples and definitions. We call this second type of feedback *author feedback*. Thirdly, an author receives feedback from a group of learners who study a course. With this information the author will be able to optimise his course. We call this third type of feedback *group feedback*.

3. PROPOSED RESEARCH

In this project we try to develop generic, i.e. domain and task independent, feedback mechanisms that produce semantically rich feedback to learners and authors during learning and authoring.

We imagine an eLearning environment in the computer science field, 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 course development. To be able to produce semantically rich feedback the system should contain several types of knowledge. At this moment, we distinguish knowledge about:

- **Domain** – For example Communication Technology
- **Model language** – For example UML/OCL or CC
- **Task/method** – The phases during design in which an artefact is build
- **Education/didactics** – For example concept learning, problem solving, examples and definitions.
- **Feedback** – For example different types of feedback and patterns/phases in dialogs.

To represent this knowledge we make use of ontologies. For a short explanation of ontologies see appendix C. In this project we mean with an ontology [19]: (1) a vocabulary: an informal description of the concepts, predicates, functions and constants of the domain of interest, and (2) an encoding of general knowledge about that domain in terms of axioms. With the axioms we describe the domain precisely and we are able to run inference procedures to automatically derive consequences (in our case to produce feedback).

More formally, we define an ontology O as a mapping (M), or translation, from a domain of interest (D) to a semantic domain (D^s):

Ontology $O = (D, M)$, where $M :: D \rightarrow D^s$

Figure 2 gives an overview of the central idea of a generic feedback mechanism:

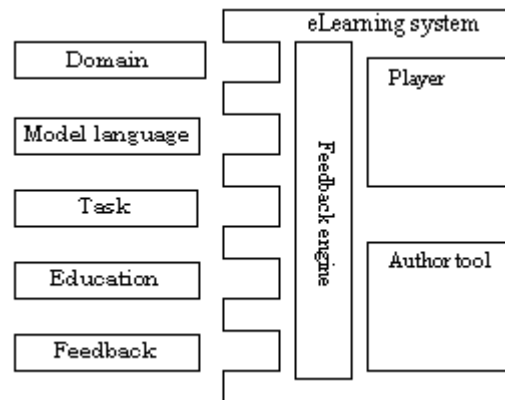


Figure 2. Functional architecture

The eLearning environment consists of four main parts: a learner site, an author site, a feedback engine and a set of ontologies as pluggable components. The learner site consists of a design and learning environment in which a learner can learn concepts, construct artefacts and solve problems. The author site consists of an authoring environment where the author develops and maintains courses and course related materials like ontologies, examples and feedback. The feedback engine produces feedback to learners as well as to authors.

The feedback engine produces generic feedback and specific feedback. For this purpose the feedback engine uses the five ontologies mentioned, which can be plugged in: they are the arguments of the feedback engine. Generic feedback is independent of the ontologies used and is applicable to all design activities and artefacts. Changing for example the domain ontology doesn't mean adaptation of the feedback engine. Specific feedback is defined by the author and can be domain, model language or task specific.

Example of learning site 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 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 on the basis of the feedback ontology, about the completeness and correctness of the list of concept and relations.

In a second step the learner is asked to design a part of a local area network (LAN), using the network model. 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.

Example of author site The author develops and optimises a course. He/she has to choose, develop and/or adapt particular ontologies and develop related material like examples, definitions, etc.. The author gets feedback, based on an integral analysis of the domain, education and feedback ontologies, 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.
- **Recursive concepts:** A concept can be defined in terms of itself. Recursive concepts are often undesirable. If a concept is recursive, there should be a base case that is not recursive.
- **Synonyms:** Concepts with different names may have exactly the same definition.
- **Homonyms:** A concept may have multiple, different definitions.

Furthermore, the author receives group feedback to be able to optimise the course material.

3.1. Research questions

The two main research questions in this project are:

- (1) How can we construct an eLearning system, which produces semantically rich feedback to
 - a. A learner during design tasks
 - b. 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 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?
- Which aspects are domain specific and which dependencies exist between the different ontologies?

Finally, we want to realise such a system as a prototype with one or two instantiations, for example: domain ontology 'Communication Technology' and modelling languages 'UML/OCL, CC and Petri-nets. We are thinking about a stress test in another domain, for example JAVA programming.

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

3.2. Related work

There exists a lot of literature about feedback from a psychological and educational view. Mory [17] has produced an extensive overview of feedback in the field of education in which definitions, a history of feedback research, models of feedback, research variables of interest, the influence from constructivism as a new learning paradigm and the possibilities of (new) technology are discussed. She also gives a summary of learning theories as concept learning, problem solving and constructivism. Within the OTEC Hummel [13] and Martens [16] have performed feedback related research. Hummel provides guidelines for cueing with the aim to improve (1) task performance in complex learning environments, (2) schema construction and, (3) monitoring. Schemata play an important role in solving complex problems. Cueing is considered to facilitate both recall and interpretation of available schemata (task performance) and the construction of new schemata (learning). Martens has determined the value of feedback in eLearning environments and distance education. He concluded ...

An example of a design environment that produces semantically rich feedback is described by Fischer (e.a.) [11]. They have described a kitchen design environment that produces critiques: dialogs in which the interjection of a reasoned opinion about a product or action triggers further reflection on or changes to the artefact being designed. They argue that computer-based critiquing systems are most effective when they are embedded in domain-oriented design environments. These environments are knowledge-based computer systems that support designers in specifying a problem and constructing a solution. Three embedded critiquing mechanisms are implemented: generic, specific and interpretive. Unclear is how much effort is needed to realise such a system and how general the (interference) mechanisms are, i.e. what has to be done if for example the system is used in another domain or new feedback types are added.

Aroyo (e.a.) have described authoring tools based on ontologies. In [2] an approach is presented based on a layered ontological paradigm to authoring support for web-based courseware. The ontology-based layers serve as a basis for formal semantics and reasoning support in performing generic authoring

tasks. Two ontology types are used: course ontology and domain ontology. The focus is on the domain ontology creation process. Because authoring of ontologies is difficult, extra functionality in the form of layers is developed. They claim that such an extension allows intelligent authoring assistance. In [3] the automatic and semi-automatic performance and support of courseware authoring activities, including hints and recommendations, are described. The necessary knowledge representation is realized by a courseware authoring tasks ontology.

In our research we will also work on supportive mechanisms to authoring tasks, i.e. feedback as supportive mechanism to build ontologies and courseware related materials. Besides feedback to authors we will also give feedback to learners. Furthermore, we will focus on generic feedback mechanisms where ontologies are arguments of the feedback engine.

In [12] a computational mechanism is defined for the delivery of case contents, which is part of case-based design education. Procedural semantics are defined on conceptual graphs. These graphs incorporate the dynamic modelling capability of Petri-nets. A case is initially opaque to the learner. During interaction a case will be made transparent gradually via engaging the students in problem-solving activities. The idea is illustrated by solving physics problems.

In [15] an authoring system and process is described that uses ontologies to produce feedback (error, warning and suggestion) to an author. Domain ontology as well as task ontology is used. The focus in the paper is on the functional structure of such an authoring system and the generation of messages using axioms defined in the ontology. The ontologies are enriched with axioms. On the basis of the axioms, messages of various kinds can be generated when authors violate certain constraints. Examples are checks on consistency and completeness. Templates are used to define the axioms, which consist of four portions: name, participants, axiom body, type.

We want to use the same type of knowledge representation and feedback generation. However, extensions are the separation of the common feedback engine, which produces feedback, and the ontologies, which are arguments of the engine and can be changed. Besides domain an task ontology we want use more ontologies to enrich the feedback. Furthermore, besides feedback to the author we want produce feedback to the learner.

3.3. Type of research

In this research we will develop a prototype of a design oriented eLearning system that produces automatically semantically rich feedback. The focus will be on knowledge structuring (for example by using grammars) and analyse techniques (for example by grammar analyse techniques and techniques from the artificial intelligence) to produce feedback.

The functionality will be developed based on examples and generalised in second instance.

The functionality developed will be realized by prototypes. Papers will found the results.

3.4. Domain and context of this research

This research will focus on design-oriented education in computer science and takes place within the faculty of Computer Science of the Open University of the Netherlands. People involved are: Johan Jeuring (professor) and Frans Mofers. This project is part of the STEL-project [14]. Furthermore, this research will be performed in cooperation with the Software Technology group of the Utrecht University. Activities related to group feedback are performed within the Alfabet project.

Other possible interesting research groups are:

- OUNL OTEC: prof. R. Koper and H. Hummel
- Information System Group, Department of Mathematics and Computer Science, University of Eindhoven: L. Aroyo
- Department of Computer Science, University of Twente: L. Aroyo

3.5. Products and schedule

We have planned the follow research activities and products during 2004/2005:

- Firstly we describe a research plan, which will be converted into a paper (May, 2004).
- At the moment we are developing structuring mechanisms and algorithms to produce feedback to the author. This will result in a prototype and a paper (fourth quarter 2004).
- Within the Alfabet project we are currently developing ideas about group feedback to the author. These will be described in a paper (July, 2004).
- As part of the second prototype of the Alfabet project we realise a prototype of our ideas. This will be described in a paper and realised by a prototype (third quarter 2004).
- At the beginning of 2005 we will develop a classification of feedback types based on a literature study. This will result in a paper (first quarter 2005).

The last subject, classification of feedback types, will influence the future activities and subjects.

The next table summarize the planned activities and products.

| Subject | Period | Product |
|--|--------|---------------------|
| Research plan | 2004 | Paper |
| Start authors feedback | 2004 | Prototype and paper |
| Group feedback: ideas (Alfabet) | 2004 | Paper |
| Group feedback: results 2e prototype | 2004 | Prototype and paper |
| Classification feedback in (design oriented) eLearning systems | 2005 | Paper |

In the first quarter of 2005 a PhD commission will be formed. Furthermore, we will look for funding (NWO).

4. LITERATURE

- [1] Antoniou, G., *The semantic web: a primer*, Vrije Universiteit Amsterdam, 2003
- [2] Aroyo, L., Cristea, A., Dicheva, D, *A layered approach towards domain authoring support*, ICAli02 (2002)
- [3] Aroyo, L., Dicheva, D., *Courseware authoring tasks ontology*, Proceedings of the international conference in education, IEEE, 2002 (2)
- [4] Boardeau, J., Mizoguchi, R., *Ontological engineering of instruction*, Artificial intelligence in Education, IOS Press, 1999
- [5] Borst, P., *Construction of engineering ontologies for knowledge sharing and reuse*, Ph. D. thesis, University of Twente, 1997
- [6] Coelho, E., Lapalme, G., *Extending Otololingua for representing control knowledge*, 1995
- [7] Dijkman, R.M., *Calculating with concepts*, Thesis, University of Twente, 2001
- [8] Dijkman, R.M., Pires, L.F., Joosten, S.M.M. *Calculating with concepts: a technique for the development of business process support*, Workshop on practical UML-Based rigorous development methods, Toronto, 2001
- [9] Distefano, J.J. , Stubberud, A.R., Williams, I.J,, *Theory and problems of feedback and control systems*, SI (metric) edition, 1988
- [10] Duffy, T.M., Cunningham, D.J. , *Constructivism: Implications for the design and delivery of instruction*, in *Handbook of research for educational communications and technology*, D.H. Jonassen (Ed.), New York: MacMillian Library Reference, 1996
- [11] Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., Sumner, T., *Embedding critics in design environments*, *The knowledge engineering review*, Vol 8:4, 1993
- [12] Fung, P.W., Kemp, R. H., *A generic graph model for case-based tutoring*, Artificial intelligence in Education, IOS Press, 1999.
- [13] Hummel, G.K, Nadolski, R.J., *Cueing for schema construction: designing problem-solving multimedia practicals*, *Contemporary educational Psychology* 27, 229 - 249, 2002.
- [14] Jeurig, J. *STEL - Software Technology for ELearning*, not publiced
- [15] L. Jin, W. Chen, Y. Hayashi, M. Ikeda, R. Mizoguchi, *An ontology-aware authoring tool*, Artificial intelligence in Education, IOS Press, 1999
- [16] Martens, R.L., *The use and effects of embedded support devices in dependent learning*, Phd thesis (OTEC), Lemma, 1998
- [17] Mory E.H., *Feedback research revisited*, in *Handbook of research for educational communications and technology*, D.H. Jonassen (Ed.), New York: MacMillian Library Reference 2003
- [18] T. Murray, S. Blessing, S. Ainsworth, *Authoring tools for advanced technology learning environments: towards cost-effective adaptive, interactive, and intelligent educational software*, Semi final drafts, <http://helios.hampshire.edu/~tjmCCS/atoolsbook/chaptersV2/ChapterList.html>, 2002
- [19] Russell, S, Norvig, P., *Artificial intelligence, A modern approach*, Prentice Hall International editions, 1995.
- [20] Stefik, M, *Introduction to knowledge systems*, Morgan Kaufmann publishers , Inc., 1995

- [21] Steels, L., Kennissystemen, Addison-Wesley, 1992.
- [22] S. Stoyanov, Mapping in the educational and training design, Phd thesis, Print Partners Ipskamp, 2001
- [23] Tosti, D., Jackson, S.F., Feedback, in Handbook of human performance technology, second edition, Jossey-Bass/Pfeiffer, 1999
- [24] Uschold, M., Gruninger, M., Ontologies: principles, methods and applications, AIAI-TR-191 (to appear in Knowledge Engineering Review, volume 11, number 2, June 1996).
- [25] J. in 't Veld, Analysis of organization problems (Dutch version), 1983, Elsevier
- [26] J. Warmerdam, . Kleppe, Praktisch UML, Addison Wesley, 1999

5. APPENDICES

Appendix A Concept learning and Problem Solving

Concept learning [17] Concepts are types of classifying rules that are used to facilitate the classification of instances through acquiring definitions, attributes and examples. Concepts can be concrete and defined. Concrete concepts represent categories determined on the basis of perceptual features, whereas defined concepts represent semantic categories that may or may not have a perceptual basis.

Concepts have both declarative and procedural components that require instruction designed to convey both of these learning outcomes. Declarative strategies help make information about the concept meaningful to the learner, and procedural strategies produce accuracy and ease in performance in concept classifications skills. Concept knowledge is more than just the storage of declarative (or verbal information) knowledge, embodying also an understanding of a concept operational structure within itself and of structure between associated concepts. Because conceptual knowledge is the storage and integration of information, and procedural knowledge is the retrieval of knowledge in the service of solving problems, instruction could typically include portions that focus on verbal information outcomes (the declarative component) and intellectual skill (concept) outcomes (the procedural component).

Using computers, concepts can be effectively taught and tested using schema representations and mapping techniques [22]. The primary method of teaching concepts usually involves presenting a definition or classification rule, followed by sets of examples and non-examples. The test whether a concept has been learned is to present the student with new instances of the concept not previously encountered to see if the student could classify the instances correctly [17]. A good introduction to schema theory as part of the cognitive psychology can be found in [10]

In our opinion this way of concept learning and testing can be, especially in the field of computer science education, extended with the use of formal representations and languages (like UML/OCL and CC). Using formal representations and languages the student gets not only a *deeper* understanding of concepts structures and the structure between associated concepts, but acquire also practical experience with concept modelling. These form a necessary and valuable basis for problem solving activities.

Problem solving [17] In the domain of problem solving, a learner must select and combine multiple rules to reach a solution. This may require that learners use declarative knowledge and cognitive strategies within a content domain and combine previously learned relational and procedural rules to solve a previously unencountered problem. The following stages often occur during a problem-solving task, and not necessary in the same sequence:

- (1) Clarify the given state including any obstacles or constraints.
- (2) Clarify the goal state, including criteria for knowing when the goal is reached

- (3) Search for relevant prior knowledge of declarative, rule, or cognitive strategies that will aid in solution.
- (4) Decompose the problem in sub problems with sub goals
- (5) Determine a sequence for attacking sub problems
- (6) Consider possible solution paths to each sub problem using related prior knowledge
- (7) Select solution path and apply production knowledge (rules) in appropriate order.
- (8) Evaluate to determine if the goal is achieved. if not revise by returning to (1) above.

Design The terms design and problem solving have been used interchangeably in the literature about educational design and engineering design. The term 'problem solving design' is also used and might be interpreted at least in three ways: (1) solving design problems (focuses on the specifics of design problems), (2) problem solving in the design process (focus on process characteristics of problem solving) and (3) designing problem solutions (focuses on tools, procedures and techniques to produce deliverables) [22].

Appendix B Feedback and system theory

System theory There exist much literature about feedback and control systems. Feedback can be defined as 'that property of a closed-loop system which permits the output (or some other controlled variable of the system) to be compared with the input to the system (or an input to some other internally situated component or subsystem of systems) so that the appropriate control action may be formed as some function of the output and input' [7]. For descriptions and definitions about terms like system, loop, ..., etc., see for example [7] and [25].

Four methods can be distinguished to affect a system[25]

- (1) **Steering or arrange** which assumes a stable process without any feedback and no internal or external disturbances.

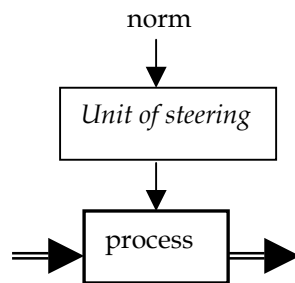


Figure 3. Steering or arrange

- (2) **Feed forward systems** in which the process input is affected to create a stable process output. The assumption is that the noise is known and can be measured. The noise determines the intervention.

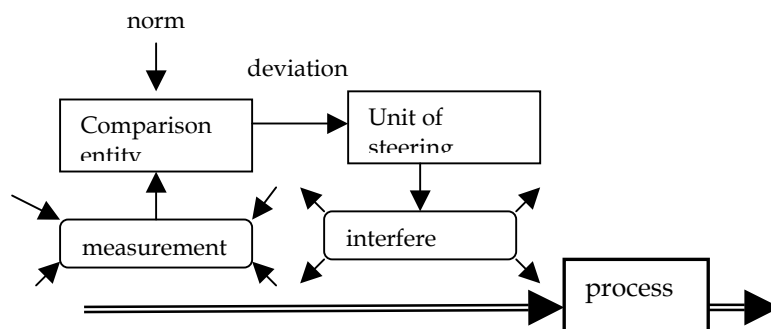


Figure 4. Feed forward

- (3) **Feed backward systems** in which the process output is used to adjust the process. The system output (the result) determines the intervention. The noise could be unknowable.

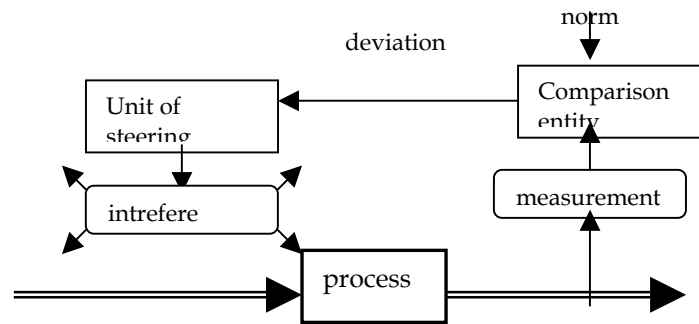


Figure 5. Feed backward

(4) **Addition of what is missing** completion of what was expected.

Some examples in the field of education are:

- (1) A static eLearning system in which content and structure is fixed.
- (2) An eLearning system which selects a (fixed) study path on the basis of an assessment.
- (3) A eLearning system where the learning content, routes and presentation are adapted during study time.
- (4) Based on tests a student is forced to study additional tasks as a result of a summative test.

In first resort our focus is on (3), where learner actions forms the measurement and are compared with norms.

Appendix C Ontologies

There exists a lot of literature about ontologies and the use of ontologies in the field of education. General introduction texts are [1][5][19][20][21] and [24].

Definitions of ontologies Several definitions can be found in the literature. Uschold and Gruninger defines an ontology as ‘a shared understanding of some domain of interest, which may be used as a unifying framework to solve problems. An ontology is a more explicit conceptualisation of the domain of interest’ [24]. Borst [5] defines ontology as ‘an explicit specification and conceptualisation’ and remark that to be able to understand this definition it must be clear what a conceptualisation is. A conceptualisation is a structured interpretation of a part of the world that people use to think and communicate about the world. So, ‘ontology is a formal specification of a shared conceptualisation’. This definition emphasizes the fact that there must be agreement on the conceptualisation that is specified. The reason for including this is that the ability to reuse ontologies will be almost nil when the conceptualisation it specifies is not generally accepted.

Parts and levels of ontology Ontologies specify the objects in the domain of interest and which characteristics, attributes, roles and relations are of importance. The list of objects is recited in a vocabulary: a list of terms and specification of their meaning (i.e. definitions). The degree of formality of ontologies varies: from highly informal (glossary) to rigorously formal (first order logic with a rigorously defined syntax and semantics) [21][24]. Boardenau and Mizoguchi talk about ontologies of three levels: (1) structured collection of terms, inclusive a ‘is-a’ hierarchy, (2) adds of formal definitions and necessary relations and constraints as a set of axioms (declarative and formal), and (3) ontologies become executable [4]. We think that a second layering mechanism could be useful. During course and ontology development in the field of Communication Technology we experience a layering of (1) fundamentals, (2) architecture and (3) applications is valuable. The lowest layer consists of stable concepts like mathematical and physical issues (for example Fourier transformation and Theory of Shannon). They may assume to be unchanged for long time. Architecture concepts make use of fundamental concepts and are less stable. Here techniques like Internet protocols are defined, which are useful for certain years. Applications make use of architecture concepts. Concepts like Bluetooth are represented. Because of changing underlying concepts, these concepts are short stable.

Types of ontologies Related to education Antoniou distinguishes models for content, pedagogy and structure [1]. Coelho (e.a) distinguish domain and control knowledge and stress this distinction as bases for reusability [6]. Boardeau (e.a.) make distinction between (1) task ontology: that characterizes the computational architecture of a knowledge based system which performs a task, and (2) domain ontology: characterizes the domain knowledge where the task is performed, such as diagnosis, planning. Instruction is a task supporting the learning process [4]. In the field of knowledge based systems the use of different models (like domain and casus models) is known [21].

Building and implementing ontologies For reuse of ontologies it is essential that they are formally specified. Much formalisms can be used. Examples are first-order predicate logic, Ontolingua, MODEL and CML. Most specification languages are based on predicate logic or meta logic. The syntax of Ontolingua

definitions is based on a standard notation and semantics for predicate calculus called Knowledge Interchange Format (KIF), a monotonic first order logic that has been slightly extended to support reasoning about relations. KIF is intended as a language for the publication and communication of knowledge [5][24]. In Ontolingua an ontology is called a theory, a term borrowed from logic. A theory consists of definitions of classes, relations, functions,, class instances and axioms. See for Ontolingua: <http://ontolingua.stanford.edu>.

The structure of an ontology can be presented by a OMT conceptual schema, which gives an overview of the concepts and the relations in an ontology. Conceptual schemata cannot replace an ontology because of their expressiveness it too limited. Other representations are XML, RDF, RDF-Schema and ontology languages based on logic [1]

Useful ontologies will be big. To build and manage big ontologies techniques like divide (on basis of natural viewpoints – for example biology: botany and zoology), abstraction and method (the way domain knowledge can be used to perform tasks) can be used.

As said before, it is important in the design of knowledge based systems to represent domain knowledge (the set of concepts and the relations between them) separately from control knowledge (the set of operations over the domain to accomplish a task). An extension of Ontolingua to represent declaratively control knowledge so that both domain and control knowledge can be represented using the same formalism and be ported to different representation languages is described in [6].