An Ontology as domain model in a web-based educational System for Prolog

Christoph Peylo and Wilfried Teiken and Claus Rollinger and Helmar Gust
Institute for Semantic Information Processing
University of Osnabrück, Germany
{Christoph.Peylo,Wilfried.Teiken,Claus.Rollinger,Helmar.Gust}@uos.de

Abstract
In the first section we give a very short survey on current research on web-based educational systems and related problems. In the second section we argue that knowledge representation and ontologies may offer solutions to basic problems in this area. Since a well-founded system of concepts, i.e. an ontology, will significantly advance knowledge sharing and interoperability. This will enable both the design of reusable functional components and the design of authoring tools. In the third part we introduce and discuss our approach of organising system knowledge in an ontology. Finally, we propose an approach how ontologies and KNOWLEDGE SPACE may be combined for improving both user modeling and intelligent problem solving support.

Introduction
Currently, web-based educational systems are a challenging research and developing area. Benefits of web-based education are independence of teaching and learning with respect to time and space. Courseware installed and maintained in one place may be used by a huge number of users all over the world. Albeit, most of the present systems are merely more than a network of static hypertext pages. A challenging research goal is the development of advanced web-based educational systems that can offer some amount of adaptivity and intelligence (Brusilovsky, 1999). Hence, there are some basic problems that hinder the development of such systems (Mizoguchi & Bourdeau, 1999). Since there are few reusable functional components available these systems are typically built from scratch, which makes them quite expensive. Representation of domain knowledge is often achieved in an author-dependent and intuitive way due to the lack of agreed standards. Thus, knowledge representation technology is the bottleneck for the development of any kind of advanced intelligent educational systems (Wenger, 1987).

The Role of Knowledge in Advanced Educational Systems
In the next paragraphs we present two examples, which show evidence that current trends in educational system design are requiring elaborate models of domain knowledge to enable knowledge transfer and communication.

Knowledge Negotiation Andriessen & Sandberg (1999) argue that the paradigm of Intelligent Tutoring Systems (ITS) has been replaced by other paradigms that stress the importance of learning how to learn instead of merely learning domain facts and rules of application (cf. Schulmeister (1997) and Wenger (1987)). The authors distinguish several scenarios (transmission, studio and negotiation) that reflect different educational settings and strategies. The transmission scenario is the most basic, where students have to learn facts and skills. The studio scenario leaves the responsibility of the learning process with the student, although the learning goal is fixed it may be reached by different ways. Learning in the negotiation scenario is learning to produce and comprehend discourse. Thus, this development moves away from tutoring procedural tasks and brings conceptual understanding, i.e. being able to reason about domain concepts and their relations, into focus. In this point of view a computer takes the part of a cognitive tool enabling interaction and discussion (Andriessen & Sandberg, 1999).

Web-based Educational Systems Web-based adaptive and intelligent systems (AIES) inherit their basic technologies from two kind of earlier systems: ITS and adaptive hypermedia systems (Brusilovsky, 1999). The ITS-related facilities include curriculum sequencing, intelligent analysis of student’s solutions and interactive problem solving support. Adaptive navigation support and adaptive presentation are related to the adaptive hypermedia system part.
Whereas almost no ITS systems includes educational material itself (Brusilovsky, 1999) situation is different in the context of web-based education. Knowledge which used to be taught in a more or less implicit way in traditional courses has to be made explicit. Furthermore, in traditional lecturing topics may be introduced and taught in a certain order depending on the background knowledge of the students, which is roughly the same in a certain class. Since learning by using the Internet is a more individualized learning, all the knowledge, which is relevant to understand a distinct domain, has to be organized in a way which provides the student with a comprehensive model of that domain (Edelmann, 1996). Thus, curriculum sequencing technology becomes very important to guide the student through the hyperspace of available educational material.

The focus of the first example was on new educational goals and in the second example on new educational settings. Although there are many differences between these research areas, the way domain knowledge is organized and represented is a vital issue for both of them. The importance of knowledge representation in the context of web-based AIES results from the need of curriculum sequencing and problem solving support. Both technologies address the basic principle of every intelligent educational system: it has to know the taught material (Lelouche, 1999), i.e. the domain. To enable communication between system and learner at content-level the domain model of the system has to be adequate with respect to inferences and relations of domain entities with the mental domain model of a human expert. Thus, they have to have a shared ontology.

On Knowledge and Ontology

In classic AI knowledge is defined in a strictly functional way (Newell, 1982). The relevant evaluative criterion for a knowledge base thus conceived is not truth but functional utility (Guarino, 1998). In the preceding section we provided arguments that in educational context domain knowledge has to be organized in a way that can express relationships about concepts, i.e. entities, in the domain of interest. Thus, we need ontologies to model domains in a way that they can be utilized for educational purposes.

There is a wide variety of different notions and definitions of ontologies available (cf. Wielinga & Schreiber, 1993, Gruber, 1993, Guarino, 1998, Mizoguchi, 1998). We will use the term ontology according to the definition of Sowa (Sowa, 1999):

**Definition 1 (Ontology)** The subject of ontology is the study of the categories of things that exist or may exist in some domain. The product of such a study, called an ontology, is a catalog of the type of things that are assumed to exist in the domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D.

Since our goal is the design of well-defined interchangeable components, we have chosen for our purpose a formal ontology that is specified by a collection of names for concepts and relation types organized in a partial ordering by the type-subtype relation (cf. Blaßius, Heddstick, & Rollinger, 1990). The lattices are derived from the attributes by the method of formal concept analysis (Ganter & R. Wille, 1996).

A Prolog Ontology for an Educational System

Our research on ontologies is related to the Virtual Campus Project, which is a joint project of the universities of Hannover, Osnabrück and Hildesheim. Its aim is to integrate Internet technologies into education and to develop lectures and environments for several courses accessible through the Internet. In the course of the project an educational system for PROLOG, the Virtual Campus Prolog Tutor, has been developed by the Institute for Semantic Information Processing at the University of Osnabrück. In earlier papers we gave an survey on the system and its components (Gust et al., 1999), (Peylo et al., 1999), in this paper we will focus on the knowledge representation aspects.

Our approach to represent facts, knowledge about the relationship between entities, and knowledge about actions in a single ontology is different from other approaches in this field, since we do not distinguish between task (Mizoguchi, Sinista, & Ikeda, 1996), domain, core or top ontologies (Breuer, Muntjewerff, & Breedeweg, 1999). The reason for our approach is twofold: firstly, since there is no agreed definition of "ontology" so far, we do not consider further differentiations helpful that presuppose an ontological founded hierarchy, which does not exist. Secondly, according to our definition an ontology is to represent all categories of interest in a domain, no matter whether they are natural kinds, artificial objects, events, processes or actions. Thus, the ontology has to be powerful enough to express those different categories. For structurizing a large scale ontology we propose an automatic approach like the knowledge packet structure (Wachsmuth, 1989). We will not tackle this problem any further here and leave this problem to a future paper.

Domain Model

**Theoretical background** We employ the methods of formal concept analysis (Ganter & R. Wille, 1996). Formal concept analysis is a mathematical approach, which defines a concept by its extension and intension. The extension contains all objects that belong to the concept. The intension contains all attributes that are shared by all these objects. This allows the specification of a formal concept that is derived from a formal context:
Definition 2 (Formal Context) A formal context is a triple $K = \{G, M, I\}$ where $G$ is a set of objects and $M$ is a set of attributes and $I$ is a relation. A relation $g \in m$ is read “object $g$ has attribute $m$”. Relation $I$ induces a partial order on the set of concepts, i.e. $G$.

The set of all concepts of context $K$ together with an order relation forms a complete lattice that is called the concept lattice of $K$.

Entities in the Prolog Domain du Boulay & Shothcott (1987) identified three basic fields for learning how to program:

- Syntax and Semantic of a programming language,
- strategic and tactical knowledge,
- knowledge about how to use the programming environment.

A web-based educational system has to take the concepts of each of these fields into account. Thus, the entities in our ontology are related to concepts and skills which are fundamental for understanding and learning to program PROLOG. For example, concepts like terms, facts or clauses are related to the syntax of PROLOG. Recursion and unification are basic for understanding the semantic of prolog. Whereas making use of goal order is an application of tactical knowledge and abilities like using an editor or compiling source code is part of pragmatic knowledge.

The Prolog Ontology Each entity in the ontology is described by a set of attributes (cf.) that may be interpreted as monadic predicates. Attributes may refer to the location of entities in time and space, to the relationship to other entities, to actions or to specific domain concepts. An attribute in this approach is something which is used to express a certain property or an aspect of an entity. Thus, they play the role of categories, since they are used to classify the entities, e.g. with respect to their position in time, space and relationship to other entities. Since all attributes have to be expressed in a formal and unambiguous way to enable reuse and interchange, we employ the LAMBDA notation (Curch, 1941) for defining attributes.

A short example shall illustrate how attributes, such as independent, relative and mediating (cf. Sowa (1999)), are defined:

independent $= (\forall x)(x:Entity) \land (\exists y:Entity) \neg(\Box(x \rightarrow y) \land \neg \text{partOf}(x, y) \land \neg \Box y)$. I.e., an entity is independent if and only if it is not strong rigid dependent.

relative $= (\forall x)(x:Entity) \land (\exists y:Entity) \Box(x \rightarrow y) \land x \neq y \land \neg \Box y$. This includes the case of essential parts: if an object cannot exist unless another object exists and is part of it.

mediating An Entity that brings other entities into a relationship. Formally, a relation satisfies the following axioms:

- A n-ary relation is a set of n-tupels.
- All sets of n-tupels satisfying a certain n-ary predicate define a relation, and every n-ary relation $R$ defines the predicate $\text{belongs_to}_R$.
- A n-ary relation $g \subseteq (X_1, X_2, X_3 \cdots X_{n-1}, Y)$ is called a function, if for each $x_1 \in X_1, x_2 \in X_2 \cdots x_{n-1} \in X_{n-1}$ there is a unique $y \in Y$.

The attributes used to categorize entities refer themselves to a certain conceptualization which is no explicit part of the ontology, i.e. the existence of predicates like partOf. Thus, the attributes are not founded in the PROLOG ontology itself, although they could be defined in a kind of meta ontology, of course. In our point of view it is not possible to design an ontology which contains everything in itself.

The number of attributes used to describe a domain depends on both the interactions and relationships to be expressed and the kind of objects that are to be discernible in that domain. I.e. the underlying concept of knowledge about some reality or domain is connected to the ability to discern phenomena, processes, objects, etc. (cf. Pawlak (1991)).

We give a short excerpt of our PROLOG ontology as an example:
Rule: A rule is a proposition that specifies how a situation can be transformed in a subsequent state. This is expressed in the following notation:

\[ \text{rule: proposition relationship situation subsequent.state;} \]

Inference.procedure: An inference procedure is an abstract description of a rule-based procedure for drawing conclusions. It contains rules and operations for manipulating symbols. In the above notation:

\[ \text{form script % abstract description proposition relationship situation subsequent.state % rule} \]
\[ \text{event relationship passiv symbol; % symbol manipulation} \]

Inductive.conclusion: Induction is an inference procedure that uses a finite set of rules, which can be used repeatedly:

\[ \text{form script % abstract description proposition relationship situation subsequent.state % rule} \]
\[ \text{event relationship passiv symbol % symbol manipulation} \]
\[ \text{recursion; % property of induction} \]

Note that each concept \( c \) in our example (\( c \in \{ \text{Rule, Inference.procedure, Inductive.Conclusion} \} \) is an element of the set \( G \) of objects of the formal context, e.g. \( c \in G \), and each attribute \( m \) is an element of the set of attributes \( M \) (cf. ). Attributes are inherited via the type-subtype hierarchy. Thus, the cardinality of the set of a concept's attributes induces a partial order on concepts of an ontological category.

Modelling domain knowledge in a concept lattice, as displayed in figure 1, ensures that the system "knows" the dependencies between concepts which is a prerequisite for both sequencing technologies and negotiation scenarios.

Ontologies and Knowledge Spaces

To improve intelligent problem solving support we currently work to integrate the concept of KNOWLEDGE SPACE (Falmagne et al., 1990). (Doignon & Falmagne, 1999) in our approach. According to this theory the knowledge state of a person is represented by the set of questions in the domain she or he is able to answer. Generally, an individual's knowledge state has to be inferred from the responses of questions, since it is not directly observable.

Definition 3 (Knowledge Space) A knowledge structure \( S \) that consists out of several knowledge states is called a KNOWLEDGE SPACE if \( K_i \) and \( K_j \) are any two states in \( S \), then \( K_i \cup K_j \) is also a state, i.e. \( S \) is closed under union.

Falmagne et al. (1990), Doignon & Falmagne (1999) associate with each question a set of skills, which are useful or instrumental to solve the problem. Skills may consist in knowing the intension and extension of a concept, in applying a concept to a task, or algorithms, methods, tricks etc. Dünstsch & Gediga (1995) have employed this approach for user modelling. It may be adopted to the needs of problem solving support as well. The main point here is to associate with each field that plays a role in learning a programming language an adequate set of skills. The work of Gregg-Harrison (1991) may here serve as an example: he identified fourteen basic schemata that capture the majority of simple recursive list processing PROLOG programs. The schemata in his approach can be identified with skills in Falmagne's.

Thus, combining a domain ontology with the KNOWLEDGE SPACE approach may result in our opinion in an educational system that is based on a comprehensive domain model in respect of both knowledge on the domain concepts, i.e. theoretical knowledge, and knowledge on the skills that are required in the domain, i.e. practical knowledge. This improves the intelligent problem solving support facilities, since the system may identify more precisely the skills a learner has problems with.

Conclusion

We argued that ontologies play vital role in solving current problems in intelligent educational system design. We have shown how the knowledge of a complex domain like the teaching of a programming language may be organized in an ontology. We outlined how this approach may be connected with KNOWLEDGE SPACE and used for intelligent problem solving support and modelling user knowledge in terms of skills and concepts.

We hope that the ongoing research in establishing a set of agreed ontological categories will advance to facilitate knowledge sharing and interoperability between systems. This will enabling both the design of reusable functional components and the design of authoring tools and flourish the development of educational systems.

References


