Ontologies for Education: Templates for intersecting interests

Dan Rochowiak
Cognitive Science
drochowi@cs.uah.edu
The University of Alabama in Huntsville

1. Ontologies and education: the debate

The traditional objectivist view of knowledge — represented all too often in actual classroom practice (Marzano, 1988) — holds that students receive knowledge from the teacher. To demonstrate what they learn, students reproduce information on tests, rather than undertaking actual performances. According to some observers (Beyer, 1987; Jones, Palincsar, Ogle, & Carr 1987; Resnick, 1989), classroom practice that adheres to this view accounts for much of the failure to improve the ability to think critically or constructively about problems and information.

Although developing the thinking ability has long been the stated goal of schooling, educators did not begin to attend seriously to the teaching of thinking until the 1980s (Worsham and Stockton, 1986). "Constructivism" is a new theory of learning that is presently receiving much attention as an alternative to the traditional view of knowledge (Resnick, 1989). Constructivism acknowledges three principles of learning:

1. Learning is a process of knowledge construction, not of absorbing and recording pieces of separate information.
2. Learning depends on previous knowledge as the principal means of constructing new knowledge.
3. Learning is closely related to the situation or context in which it takes place.

Four common findings from research about thinking accord well with practice in education (Resnick & Klopfer, 1989):

1. Knowledge and expertise are the foundations for thinking and learning about a particular domain.
2. The disposition to use skills and knowledge, as well as to possess them, is part of learning.
3. Social communities play a key role in developing thinking abilities.
4. Apprenticeships are powerful frameworks for learning.

Many education researchers now accept the definition of thinking as a search for meaning, involving the mental processes that make sense out of experience. As a motto, learning is thinking (Jones et al., 1987). That is, learning depends on prior knowledge and the specific mental strategies that evoke understanding in perception, prior experience, conscious manipulation, incubation, and intuition.

The literature on thinking covers a wide range of topics. Resnick (1987) described six broad categories of thinking skills:

1. problem solving in the disciplines,
2. general problem solving,
3. reading and study strategies,
4. self-monitoring,
5. components of intelligence, and
6. informal logic and critical thinking.

Marzano and colleagues (1988) cite similar categories, but these authors also note that advocates of thinking skills offer a "bewildering" assortment of strategies for teaching such skills. Costa (1991), for instance, considers programs from more than 40 strategies.

Selecting a program is difficult and as we will show carries a variety of ontological commitments. Should teachers use the traditional, objectivist view of learning? If not, on what basis do they select from among the alternatives? Do they construct their own programs? As a matter of practice, teachers must discover which theoretical bases are most appropriate for their students, the settings in which they teach, and their own teaching styles. If the emerging literature on thinking is correct in its claims about learning, teachers will, after all, eventually use what they learn to construct their own instructional models, ontologies, and routines.

2. From educational epistemology to ontology

Let us suppose that we intend to build computer systems that tutors, instructs or teaches students. Let us further suppose that we can adopt either the objectivist or constructivist models. If the objectivist position were accepted, then we would imagine the transfer of ideas from the teacher to the student in instruction and from the student to the teacher in testing. These ideas would amount to a description of the characteristics and behaviors of the things in the domain. In this sense what would be transferred is the ontology of the domain. It should be noticed that this ontology is taken as independent of both the teacher and the student. The case is vastly different, and perhaps more interesting, if one takes the constructivist turn. This is the turn that we shall take.

Ontology can be understood as either the descriptions about some domain or as the specification of the things that make up a domain. Although there are quite a few philosophical problems about these alternatives, we will, as good constructivists, simply assume that an ontology is a collection of descriptions about a domain. This collection of descriptions generally includes elements that provide categories, measures, and criteria, specify ways in which place, time, and change are specified, indicate the ways in which things can be composed and decomposed, and the relations among all of these and the beliefs and other mental states about them. In education all of these are important.

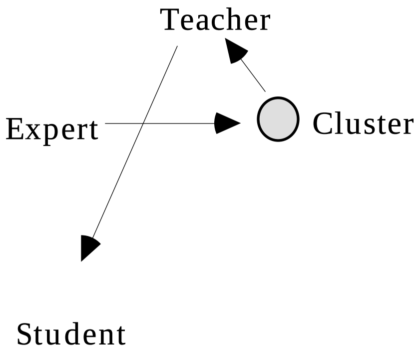
We will focus on the discovery program of learning which is consistent with constructivism. In this program there are three phases of learning. The first phase is discovery and exploration. In this phase, the student is encouraged to examine a particular domain and create whatever hypotheses and beliefs she feels adequate for understanding the domain. She is also encouraged to formulate questions about what she does not understand. In the presentation phase, the student is present with material about the domain that more adequately organizes and describes the domain. In the final phase the student applies what she has learned in the first two phases to cases, which may or may not be examples of the presented material.

3. Multiple ontologies

The introduction of a computer system into this program of education raises several ontological problems. There would appear to be at least five ontologies in the computer enhanced educational context: the domain (or expert's) ontology, the teacher's ontology, the student's ontology, the educational program's ontology, and the computer system's ontology. All of these are the part of the context in which the computer-enhanced education must exist.

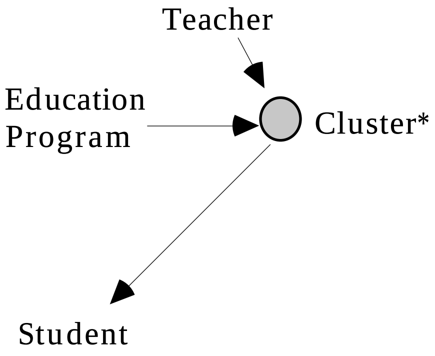
In the process of building an educational computer system there are three important ontological transitions.

In the first transition, the teacher constructs an ontology for some cluster of items in the domain from the expert's knowledge. In doing so, however, the construction is guided by the need to communicate with the student



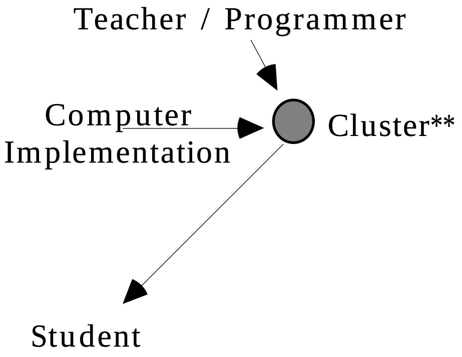
Teacher constructs
the cluster's ontology
from the expert's
knowledge for the
student

In the second transition, the teacher reconstructs the cluster's ontology in terms of the educational program's ontology.



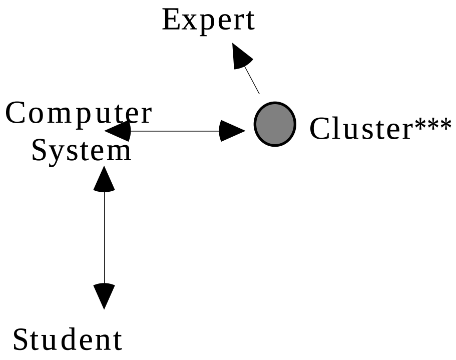
Teacher reconstructs
the cluster's ontology
(forming cluster*)
in light of the
ontology of the
educational program
for the student

In the third transition this reconstructed ontology is once again reconstructed by the implementer (ideally the teacher-programmer) into one that is appropriate to the computer tutoring system.



Teacher / programmer
reconstructs
the cluster's ontology
(forming cluster**)
in light of the ontology
of the computer system
for the student

At this point the system is complete and the student interacts with it in the attempt to reconstruct the expert's ontology. In large measure this process of reconstructing is what the constructivist refers to as thinking and the result of the process is what the objectivist understands as being transferred.



The student attempts to
reconstruct the expert's
ontology by interacting
with cluster*** as
mediated by the
computer system

Given that there are these different ontologies and accesses, is there some way in which a sufficient degree of commonalty or accommodation can be created so the a system can be designed from the various parts? Is it possible to design a sufficiently rich and acceptable ontology that educational AI systems have a chance at being designed and implemented?

Our answer is a qualified yes. The qualifications are that the degree of commonalty cannot be great and the possibility of increasing commonalty should be well understood. This is akin to the well-known edicts in rule-based or expert systems technology. For there to be a benefit from the AI approach, the problems must be difficult, but not so difficult that the solution procedure is not known.

4. Ontologies and templates

Rather than taking ontologies to be a set of fixed elements in a domain, we will take ontologies to be templates.

In advancing the notion of explanatory unification in scientific reasoning, Kitcher developed the notion of a general argument pattern. (Kitcher, 1981) A general argument pattern consists of:

1) a schematic pattern,
2) a set of filling instructions for the schematic elements in the pattern,
3) a classification describing the inferential characteristics of the schematic pattern.

This notion becomes the typical logical notion of an argument pattern in those cases where the conditions on the classification are as stringent as possible and the filling instructions are as relaxed as possible. In this sense, the general argument pattern becomes the logical argument pattern when there are no restrictions on the filling instructions and the classification is determined exclusively by the logical connectives in the schematic pattern. The notion of a general argument pattern does not preclude the production of logical argument patterns, but these are produced only under special conditions. As actual conditions become more stringent, general argument patterns more closely resemble the micro-theories of open systems. These micro-theories can be understood as the actual content of thinking.

A general argument pattern is a template. The filling instructions can reasonably be viewed as rule-like structures that support the notions of typical and exceptional values. These structures link the schematic terms in the template to procedures that will produce acceptable values for those terms. Thus, the filling instructions are similar to operational definitions and the names that are found in the template correspond to these definitions and can be understood as the names that when used in descriptions refer to the ontological elements of the domain.

The filling instructions do two things. First, they indicate semantically the kinds of things that the schematic variables represent. Thus, the filling instructions indicate the types of substitutions that are legitimate in a domain. In Newtonian physics, for example, these first sort of filling instruction would restrict the sorts of things that can be substituted for ‘F,’ ‘m,’ and ‘a.’ While we typically think that ‘F=ma’ means that “Force is equal to mass times acceleration” this is not obvious from the schema as it is presented. ‘F,’ ‘m,’ and ‘a’ could represent other things. Second, the filling instructions indicate how to replace the schematic terms. For example, a filling instruction may specify that ‘a’ is to be replaced by a function of an object's coordinates and its time derivatives. Or again, the filling instruction might simply allow that 'a' should be replaced by the reading of some instrument or that the value of 'a" is in a particular field of a screen display. Further, the filling instruction might specify a computation or action that should be perform by the computer system or the user. In each of these cases, the filling instructions are taken in terms of a local ontology. Different ontologies may have different filling instructions. The local ontology will be a function of the expert's, teacher's, and student's assumed or proscribed ontological commitments and the phase of the educational program. If the student is taken as the driving force in the learning, then the filling instructions addressed in the representation will change over time and the ontological commitments of the student and the computer system will likewise change.

5. A 'toy' example

We will now develop a 'toy' example. This is a 'toy' example since it does not present all of the detail that a real example would. The toy example focuses on the idea of momentum. From the teacher's and the student's point of view the system is composed of lessons and each lesson has a discovery, presentation, and application phase. From the domain expert's point of view these phases don't exist, and would not be part of the domain expert's ontology. Each phase must access the momentum template that was constructed by the teacher-programmer on the basis of what the expert had presented. The template might look something like this:

```
elements:
      m, v, object, direction
principle:
      for all objects
            the sum of m * v  for each object before equals
                  the sum of m * v  for each object after
m:
      if phase is discovery, then perform m_demo
      if phase is presentation, then perform m_presentation
      if phase is application, then get value
v:
      if phase is discovery, then perform m_demo
      if phase is presentation, then perform v_presentation
      if phase is application, then get value
direction:
      if phase is discovery, then ignore
      if phase is presentation, then ignore
      if phase is application, then get value
object:
      if phase is discovery, then ignore
      if phase is presentation, then perform o_presentation
      if phase is application, then get value
student:
      beginning, experienced
```

The template for the expert would elaborate the elements and principles parts of the template, and the student template would elaborate the instruction_types template. Thus there would be three templates for the momentum ontology. From the expert's point of view the ontology would be composed of those characteristics and relations demanded by a Newtonian analysis of motion. This analysis may have many more elements that the teacher template. For example, the expert template may also have elements for friction and spin. Those elements not specified are given default values. The student template would reference yet other elements. For example, there would be a student model and catalog of response gestures.

6. Conclusions

If one is constructing software for educational or instructional use, it will not be surprising that various ontologies are encountered about what is in one sense the same domain. Considering the general domain of physical objects in motion and in particular the momentum domain cluster, one would expect to find that in constructing the lesson the teacher will not expose the student to a complete ontological specification of the cluster. Rather certain elements with certain restrictions will be selected for the lesson. Further, there will be some elements of the domain lesson that simply do not appear in the expert or student ontology. The ontology that student "sees" is a product of the expert, teacher, and student ontology imbedded in the system.

One might think that if there were a place where knowledge is shared, it would be the educational arena. However, it is this very arena that engages the difficulties in building a common ontology, a common understanding of the world. In education various ontologies come together in the effort to generate a context in which the student learns. What should be remembered in the discussions and debates over common ontologies and knowledge sharing is the constructive dimension of the enterprise.

7. References

Beyer, B. (1987). Practical strategies for the teaching of thinking. Boston, MA: Allyn and Bacon.

Caine, R., & Caine, G. (1991). Making connections: Teaching and the human brain. Alexandria, VA: Association for Supervision and Curriculum Development.

Fikes, R., & Kehler, T. (1985). The role of frame-based representation in reasoning. *Communications of the ACM*, **28**(9), 904-920.

Gruber, T. R. (1991). The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In J. A. Allen, R. Fikes, & E. Sandewall (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference,* Cambridge, MA, pages 601-602, Morgan Kaufmann.

Guarino, N. (1992). Concepts, Attributes, and Arbitrary Relations: Some Linguistic and Ontological Criteria for Structuring Knowledge Bases. *Data and Knowledge Engineering*, **8**.

Interrante, L. and Rochowiak, D., Sumrall, B. (1994). Scheduling of automated guided vehicles for material handling, *Computer Control of Flexible Manufacturing Systems*, S. Joshi and J. Smith. Chapman and Hall, 1994, pp. 108-141.

Jones, B., Palincsar, A., Ogle, D., & Carr, E. (1987). Strategic teaching and learning: Cognitive instruction in the content areas. Alexandria, VA: Association for Supervision and Curriculum.

Marzano, R., Brandt, R., Hughes, C.,, Jones, B., Presseisen, B., Rankin, S., & Suhor, C., (1988). Dimensions of thinking: A framework for curriculum and instruction. Alexandria, VA: Association for Supervision and Curriculum Development.

Resnick, L. (1987). Education and learning to think. Washington, DC: National Academy Press.

Resnick, L. (Ed.). (1989). Knowing, learning, and instruction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Resnick, L., & Klopfer, L. (Eds.) (1989). Toward the thinking curriculum: current cognitive research. Alexandria, VA: Association for Supervision an Curriculum Development.

Rochowiak, D. and Interrante, L. (1991) Heterogeneous knowledge based systems and situational awareness. *AAAI '91 Workshop on Cooperation Among Heterogeneous Agents.* (Anaheim, 1991).

Rochowiak, D. The evolution of silicon life (1994). *Inner Space / Outer Space*, D. Schenker, C. Hanks, and S. Kray (eds.), Southern Humanities Press, 1994, pp. 217-234.

Rochowiak, D., Messimer, S., and Rogers, J. (1994). Critiquing with Multiple Criteria: Conflict Detection and Resolution. In AAAI'94 Workshop on Models of Conflict Management in Cooperative Problem Solving (Seattle, WA 1994), p. 97-106.

Worsham, A., & Stockton, A. (1986). A model for teaching thinking skills: The inclusion process. Bloomington, IN: Phi Delta Kappa Educational Foundation.