Chapter 11
Ontology-Based Formal Modeling of the Pedagogical World: Tutor Modeling

Riichiro Mizoguchi[1], Yusuke Hayashi[1], and Jacqueline Bourdeau[2]

[1] The Institute of Scientific and Industrial Research (I.S.I.R), Osaka University
8-1, Mihogaoka, Ibaraki, Osaka, Japan, 5670047
{miz,hayashi}@ei.sanken.osaka-u.ac.jp

[2] LICEF Research Center, TÉLUQ-UQAM
100, Sherbrooke Street West, Montréal, QC, Canada H2X 3P2
jacqueline.bourdeau@licef.ca

Abstract. This chapter discusses an ontological approach to tutoring actions design as a special case of target-world modeling. Although a lot of research on the learner model has been done to improve the adaptivity of intelligent tutoring systems (ITSs), the modeling of tutoring actions has not been sufficiently investigated. The authors have been performing ontological modeling of learning/instructional theories to remedy this situation. Intelligent tutoring systems must have a good number of primitive actions to generate intelligent actions. Paying close attention to the importance of modeling tutoring actions, we have developed an ontology of learning/instructional theories, named OMNIBUS, in the ITS domain. Drawing on our long experience in ontological engineering research, this chapter discusses the modeling of tutoring actions as well as target-world modeling per se, using an example of learning/instructional actions from the OMNIBUS/SMARTIES project.

11.1 Introduction

Advanced information processing technology, and especially knowledge processing, has influenced the design of learning support systems. Advances that have played a central role in the Intelligent Tutoring System (ITS) area include applications of artificial intelligence (AI) technology to learner modeling and the generation of tutoring actions that can be adapted to the state of the learner, based on the learner model. In both of these applications, inference techniques are the key technology. However, knowledge processing has not been focused on as the main technology, despite the fact that inference cannot work without knowledge.

Intelligence is attributed to adaptive actions of a system. Such actions can only be generated by a system that has been designed to change its actions adaptively to a situation. Researchers in ITS have placed greater emphasis on building the learner model than on modeling the tutor to perform intelligent tutoring actions.

R. Nkambou et al. (Eds.): Advances in Intelligent Tutoring Systems, SCI 308, pp. 229–247.
springerlink.com

This is natural, considering that no adaptation is possible without knowing the learner's state to which the system must adapt. After considerable research on learner modeling, we now have more results on that aspect than on the modeling of tutoring actions. We believe, as we have been claiming since our vision paper (Mizoguchi and Bourdeau, 2000), that it is time to pay more attention to the importance of modeling tutoring actions.

People usually concentrate on the adaptivity of a tutoring system from the inference point of view—which is not incorrect, as far as it goes. However, researchers need to be aware that a good number of primitive actions must be prepared in advance in order to perform such adaptive actions, because systems cannot generate intelligent actions from nothing. Although it seems obvious, the reality is that quite a few researchers do not take this simple fact into consideration. Our aim is to *model* the target world, including the primitive actions that are expected to be the infrastructure underlying intelligent actions. We believe that what has been missing in ITS research is not only the modeling of tutoring actions but also *target-world modeling* per se.

We have been conducting ontological engineering in the ITS domain for years with such *target-world modeling* in mind. In fact, our claim for the importance of ontological engineering in ITS research grows out of efforts to overcome the difficulties of expert system technology and knowledge engineering, and is in the same vein as the idea of *target-world modeling.* Expert system technology played a role in redirecting AI research away from its preoccupation with form-oriented matters such as inference/reasoning, logic, and knowledge representation for toy problems, toward content-oriented research that deals seriously with real-world problems and requires sophisticated knowledge base (KB) building technology, together with practical inference on the KBs. An AI boom resulted, and a tremendous number of expert systems were built and used in industry. Despite this success, the AI boom has subsided now because of the difficulty of KB maintenance, as well as poor intelligence resulting from a shallow understanding of the domain.

Although expert system technology enables us to develop a KB specially tuned to solve a particular problem, we cannot claim that such a system "understands" the domain and the problem. Knowledge-processing techniques in expert system technology are designed for building KBs that contain rule-coded expert heuristics solely for solving particular problems, without any justification for the individual rules in the KB or any explanation about the nature of the problem and of the domain itself. This is far from the idea of *target-world modeling.*

Ontological engineering has emerged to overcome difficulties like those encountered by expert system technology. Ontological engineering helps people build domain models and expert problem-solving activities within the domain—in the real world. It helps to reveal the essential properties and structure of the target domains and describe them in a manner understandable by computers. This is the theory and technology that can be expected to achieve *target-world modeling.*

This chapter discusses tutoring action design as a special case of *target-world modeling.* Ontological engineering is exploited in order to demonstrate its power to solve this problem, taking our research experience in the OMNIBUS/SMARTIES project as an example. The theoretical details have already been published in

(Hayashi et al., 2009c). Here, we explain the philosophy underlying the modeling of learning actions as well as our policies for ontology building. The next section presents the background of this study, followed by a brief overview of our project in Section 3. Section 4 gives a detailed explanation of the process of ontology building in OMNIBUS, with eight policies. In Section 5, we introduce a next-generation authoring tool named SMARTIES in the context of these policies. Section 6 discusses how SMARTIES works, demonstrating its unique architecture (which is different from that of conventional expert systems), and ends with some concluding remarks.

11.2 Background

The aim of the OMNIBUS/SMARTIES project includes the design of learning and tutoring actions for an intelligent authoring system. OMNIBUS is an ontology of learning and instructional theories, which models learning and tutoring (instructional) actions implicitly or explicitly.

One might think that expert system technology would be a promising approach when someone is asked to build such an intelligent authoring system. He/she would try to interview experts in instructional design and/or learning theories to uncover how they design learning/instructional scenarios and on what model of the learning/instructional world these scenarios are based. After a series of successful interviews, he/she would build a rule base that nicely simulates the behavior of experts in authoring learning/instructional scenarios.

Now, we would like to pose a question about how we can evaluate such a rule-based intelligent authoring system. It will show reasonable performance when the KB is built well. But the issue here is whether we can say that the system is wise enough to understand learning and instruction or not. We believe that the wisdom lies not with the system but still with those experts who gave their knowledge to the system. This is because the system does not “know” what learning/instruction is, and because it is not built based on a good model of learning actions.

Building a truly intelligent authoring system is a really challenging goal, one worthy of the attempt. We have tackled this problem by investigating learning/instructional theories rather than expert heuristics, motivated by the strong belief that ontological engineering is more powerful than expert system technology. We have set the following seven goals for our long-term research project:

1. Building a system that “understands” a good number of learning/instructional theories;
2. Devising a function to support the generation of a learning/instructional scenario justified by learning/instructional theories;
3. Capturing prescriptive aspects of learning/instructional theories in a declarative manner;
4. Avoiding the use of expert heuristics;
5. Avoiding the use of procedural modeling in any circumstances;
6. Being able to explain the theories it has and the rationale for the learning/instructional scenarios it generates;
7. Being able to add new theories easily.

It is apparent that form-oriented AI research would not be useful here. The issue is not the reasoning but the content modeling, which conventional AI research has not yet tackled seriously. However, the situation is improving currently, thanks to the emergence of ontological engineering, which provides theories and techniques to build a fundamental model of the target world. What we must address is the modeling of learning and tutoring/instructional actions, as *target-world modeling*, that is necessary in order for a system to perform inference. Ontology engineering provides us with theories and techniques for modeling the world of interest. Technologically, its central feature is its use of a declarative description to build a model, in a computer-understandable manner, as objectively as possible. This is why we have decided to adopt ontological engineering in our project. However, while ontological engineering thus appears to be a powerful methodology that can be applied to our research, an issue remains: the modeling of prescriptive aspects of instructional theories that relies heavily on the model of learning and tutoring/instructional actions. We have decided not to use procedural modeling. How, then, can we tackle this problem?

11.3 Overview of the OMNIBUS Project

The OMNIBUS project was motivated partially by the desire to build an innovative authoring system aware of learning/instructional theories. It is also expected to be able to produce IMS-LD-compliant learning/instructional scenarios. Named SMARTIES, it satisfies all of these aforementioned requirements as well as being able to connect Learning Objects (LO) available through GLOBE[1] with the generated scenarios.

The heart of SMARTIES is *Instructional_Learning event (I_L event)* decomposition, described in 4.5. OMNIBUS prepares all concepts necessary for performing I_L event decomposition interactively with the author (designer). The prescriptive aspects of learning/instructional theories are organized as unit operations of *I_L event* decomposition. The unit should be understood as a unit of instructional strategy. *I_L event* decomposition is done by proposing candidates for decomposition by the system, and then having authors perform a selection among the candidates. The authors could also choose to input their own strategies, in which case SMARTIES proposes tutoring actions and learner's states as terms it understands by which authors can describe their strategies. This allows SMARTIES to understand the author-input strategies. In selecting among system-provided candidates, authors can blend theories to select a candidate derived from theories other than those that prompted past selections. Note here, however, that justification for such blending is not supported by theories. It must be done at the author's risk.[2]

[1] The Global Learning Objects Brokered Exchange (GLOBE), http://www.globe-info.org/

[2] A description of how SMARTIES works can be found at http://www.ei.sanken.osaka-u.ac.jp/pub/miz/AIED07 WS_Invited.pps

11.4 Building the OMNIBUS Ontology

11.4.1 Overview

The following are some of the difficulties that must be overcome in building an ontology of learning/instructional theories:

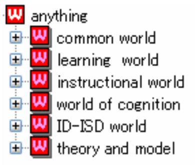
1. Defining what exactly an ontology of theories is.
2. Reconciling the, long-lasting conflicts among theoretical paradigms: Behaviorism, Cognitivism, Constructivism and Socio-constructivism have been competing for long years. It seems impossible to find a common ground on which to describe them.
3. Modeling the variety of instructional strategies in a declarative form.
4. Selecting terms/concepts to include in the ontology.

As these difficulties suggest, the project was a real challenge. We could say that capturing prescriptive aspects of theories was the hardest part of our whole enterprise. After struggling with this issue, we were fortunately able to come up with a powerful solution. The idea is a good example of knowledge transfer across domains, since the solution was devised by using a technique developed for capturing the functional structure of engineering artifacts (Kitamura, 2004). This technique will be discussed in detail below, because it has a strong potential to become a standard way of capturing human problem-solving actions in general in a declarative manner.

For the moment, let us move on to the discussion of how to build an ontology. Contrary to what most people believe, building an ontology does not mean building an *is-a* hierarchy of the target concept. What an ontology should be is an *is-a* hierarchy of all the related concepts, to capture the target concept. An *is-a* hierarchy of the target concept is just one of them. To put it in terms of our problem, the OMNIBUS ontology should comprise an *is-a* hierarchy not only of theories but also of *actions, states, events* and *strategies.*

The next biggest difficulty is reconciling the learning paradigms. We should not create new theories; we have to respect the existing theories. But this does not mean that we should blindly follow what experts claim. We as ontologists need to keep a domain-neutral attitude so as to capture reality with the highest fidelity, in order to get rid of possible non-moderate views originating in domain-specificity. In addition, we have to be empowered by sound ontological theories.

While the notion of "learning" is not shared by theorists across paradigms, we firmly believe it has to be. This is the starting point and the solid ground on which we can build an ontology of learning and instructional theories. To justify this view, let us consider a situation where we deploy learning/instructional theories in a classroom instruction situation. With what degree of accuracy can we ensure that each learner enjoys the learning conditions a particular theory requires in such a situation? Each learner is in a different state of comprehension, motivation, etc. We have to apply one instructional strategy uniformly to all the learners; that is the reality of doing education in a classroom. This clearly suggests that we have to accept approximation in deploying theories. In other words, when our interest is in how to use theories in real-world education, we can realistically expect to capture them in terms of a less precise vocabulary than those used by theorists to describe them, even though this must be unsatisfactory for the theoreticians.

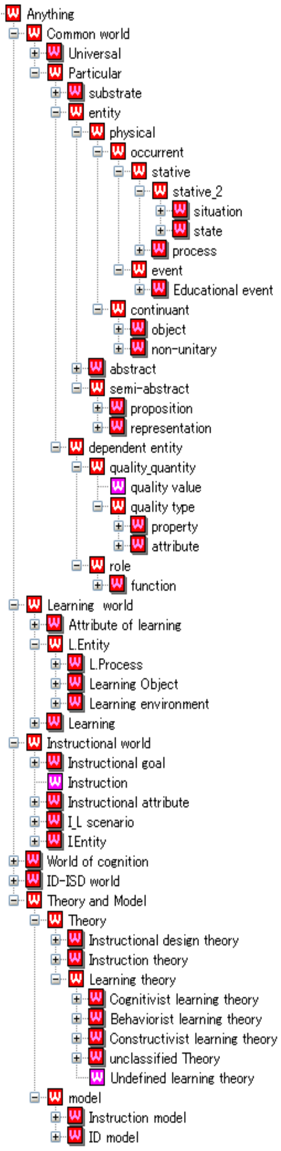


- anything
  - common world
  - learning world
  - instructional world
  - world of cognition
  - ID-ISD world
  - theory and model

Fig. 11.1 Top-level structure of OMNIBUS

Now, let us move on to the OMNIBUS ontology we have built. Figure 11.1 shows its very top-level concepts: *common world, learning world, instructional world, instructional (system) design (ID-ISD) world, world of cognition* and *theory and model.* Although it is not very compliant with the common upper ontology (BFO,[3] DOLCE[4]), it captures the learning/instructional world. In fact, we encountered the following dilemma here: If we follow the popular upper ontology for the top-level structure, then the major characteristics of the learning/instructional world disappear. If we try to make the target world more visible, then the upper-level categories become less compliant with the popular upper ontologies. We finally adopted our current approach and achieved a reasonable upper-level structure in the *common world*, in which all the common concepts are organized according to the top-level ontology YATO (Mizoguchi, 2009).[5]

Figure 11.2 shows the ontology in a bit more detail. The three worlds of *learning, instructional* and *instructional (system) design* share a common conceptual structure. *Theory and model* is organized as an independent world in which theories and models are organized in several *is-a*



- Anything
  - Common world
    - Universal
    - Particular
      - substrate
      - entity
        - physical
          - occurrent
            - stative
              - stative_2
                - situation
                - state
              - process
            - event
              - Educational event
          - continuant
            - object
            - non-unitary
        - abstract
        - semi-abstract
          - proposition
          - representation
      - dependent entity
        - quality_quantity
          - quality value
          - quality type
            - property
            - attribute
        - role
          - function
  - Learning world
    - Attribute of learning
    - L.Entity
      - L.Process
      - Learning Object
      - Learning environment
    - Learning
  - Instructional world
    - Instructional goal
    - Instruction
    - Instructional attribute
    - I_L scenario
    - I.Entity
  - World of cognition
  - ID-ISD world
  - Theory and Model
    - Theory
      - Instructional design theory
      - Instruction theory
      - Learning theory
        - Cognitivist learning theory
        - Behaviorist learning theory
        - Constructivist learning theory
        - unclassified Theory
        - Undefined learning theory
    - model
      - Instruction model
      - ID model

Fig. 11.2 More details of Omnibus

[3] Basic Formal Ontology (BFO), http://www.ifomis.org/bfo/home.

[4] DOLCE: a Descriptive Ontology for Linguistic and Cognitive Engineering, http://www.loa-cnr.it/DOLCE.html

[5] The new updated version is called YAMATO. See: http://www.ei.sanken.osaka-u.ac.jp/hozo/onto library/upperOnto.htm

hierarchies. *Common world* includes action (process), state, event, etc. These three constitute the core of our conceptualization of the learning/instructional world and will be explained in detail in the next subsection.

11.4.2 State, Action and Event

Following the conventions of AI, we represent *actions* as *state changes* between the time before and the time after the execution of the action. This is why *states* play the key role in our ontology. *Stative* is a technical term from philosophy, by which we mean "state-like thing". A *process,* which includes *actions* as its subclasses, is a kind of *stative*, but *events* are not. We have to omit the philosophical details, but *process* and *event* are clearly differentiated in philosophy (see Galton and Mizoguchi, 2009, for details). *Events* are composed of *processes* as material. The relation between the two is analogous to that between vase and clay. Like a vase, an event is a unitary whole in the temporal space, whereas a process, like an amount of clay, is not a unitary whole. Furthermore, an *action* usually plays a role in an *event* as a *context*. For example, an utterance can play an explanatory or instructional role in the context of education. This observation suggests that we should model all *processes* (actions) as *state changes* objectively (context-independently), and that they can be used to represent a context as an event.

The next topic is how to model *states*, and, in particular, how many states to model and with what degree of granularity. At first glance, it looks hard to decide the right scope and granularity for state modeling. However, the fact that our goal is to capture theories tells us that we need only model as many states as the theories require, with the requisite granularity. This policy applies to the modeling of actions as well. If we notice we need more states and/or actions when dealing with new theories, we simply add them as needed.

The above discussion can be summarized in two policies, as follows:

Policy 1: All occurrents except events are modeled in terms of state change.

Policy 2: Events are modeled using processes as material. An event is a unitary whole in the temporal space. Actions are modeled in the minimalist way to be used for event description.

State modeling is very important in OMNIBUS for the following two reasons:

1. It provides infrastructure to enable us to model all the theories on a common ground.
2. It allows us to express all phenomena occurring during learning and instructional processes, and hence it guarantees that all the application programs based on OMNIBUS work in a uniform way.

Of course, we are aware that there are quite a few people who oppose these claims. In particular, many of the theorists would like to raise counterarguments such as, "It seems almost impossible for you to implement my theory in a computer without losing the deep understanding about learning that I have put into it." We are ready to agree with such a theoretician. But, we would like to reply, "That is not the issue." What we have been trying to do is not to represent theories as accurately as possible but rather to represent them so that the computer can understand them and use them in a manner as close as possible to the way experts use and apply them in real-world education. In other words, we have been modeling theories not from the viewpoint of "making" them but from that of "using" them. Engineers seek not best but better. Their number one priority is utility rather than perfection. They make steady advances toward improving real life by using a better solution, instead of waiting for the best solution. Now, we are ready to introduce the notion of *"engineering approximation"* for modeling theories, as follows:

> *Policy 3: In order to model theories in a realistic manner, we introduce engineering approximation.*

Figure 11.3 shows an *is-a* hierarchy of *state*. *Agent state*, which represents the *state* of learner, is divided into internal and external states. The former is further divided into *cognitive process state, attitudinal state* and *progression state* and the latter into *communicative state* and *physical state*. By *communicative state*, we mean states such as *being told, having said* and so forth.

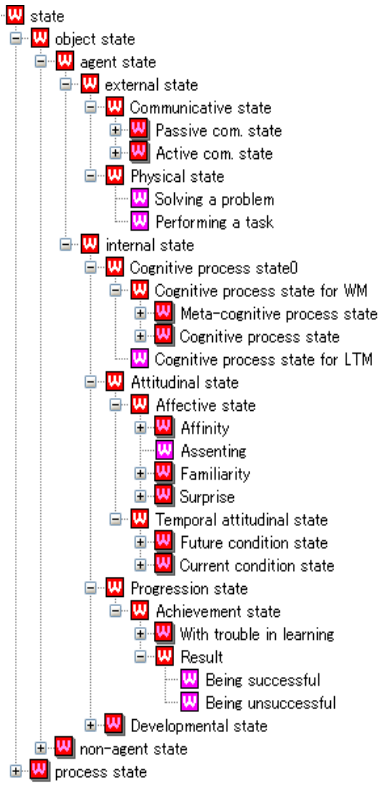


Fig. 11.3 *Is-a* hierarchy of state

11.4.3 I_L event

In order to capture prescriptive aspects of theories, we need a model of actions, and to capture actions we need a model of states. We now have both actions and states in the ontology. The next issue is how we can build models of theories in terms of actions and states. It is quite possible that such a fine-grained set of primitives would enable us to capture all the possible phenomena occurring in the course of learning/instruction, and hence it would be a good set to use. However, there remains a concern that it might be a bit too fine. If the granularity is finer than necessary, the result will be an overly complicated model. In our case, learning and instructional actions are independent of each other, so they allow us to model every possible interaction between learning and instructional actions, which yields an overly complicated model of the behavior of an authoring system. What we need is a model that can explain learning and instructional theories and use them to help build theory-compliant scenarios. In the light of this goal, the granularity should be fine enough to explain theories and coarse enough to produce easy understandable and manageable behavior on the part of the authoring system.

When we pay attention to the kinds of events, we find two primitive types: *learning event* and *instructional event*. In the former, a learner performs a *learning action* and changes his/her *state*, under some conditions which the learning theories take care of. In the latter type, on the other hand, an instructor performs an instructional action to facilitate a learning action, in order to cause a planned change in the learner. When we view these two kinds of events as a unified entity, we come up with a composite event, which we call *I_L event*, in which "I" stands for Instruction and "L" for Learning. An *I_L event* thus consists of a triple of <*instructional action, learning action, state change*>. If we can represent all the possible event sequences occurring in the course of learning/instruction, then *I_L event* can be thought of as the core of the model of prescriptive aspects of theories. The above discussion is summarized in Policy 4:

Policy 4: Try to find event units with maximal granularity, under the condition of keeping the capability of modeling all the possible phenomena under consideration.

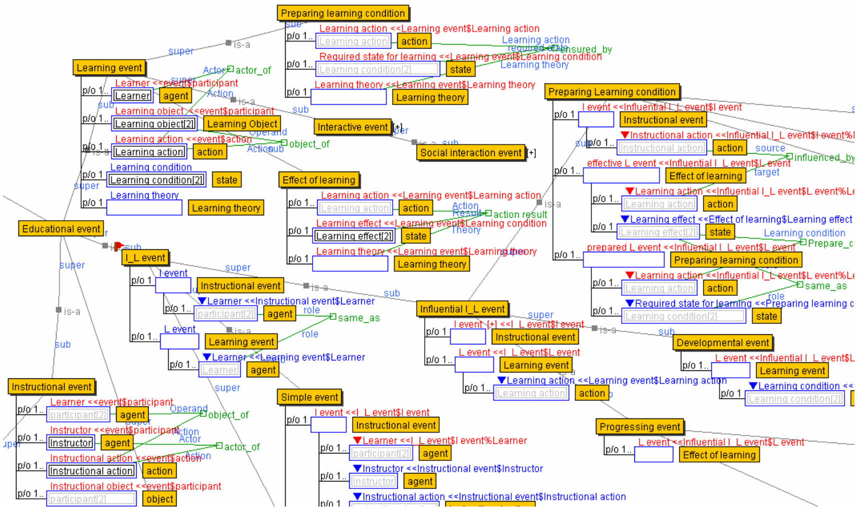


Fig. 11.4 *Is-a* hierarchy of *educational event*

What is important here can be summarized as follows: Learning theories tell us under which conditions learning happens and how learners change (what learning outcome they gain) as a result; instructional theories tell us how to facilitate such learning to maximize the learning outcome; and this nested structure of instructional and learning action is captured by the *I_L event*.

Policy 4 works well for tackling the modeling of procedural knowledge in general. It is apparent that state-based modeling works well when the modeling of actions is the main interest. As we noted above, however, we have to be careful about the granularity issue. It should be neither too small nor too large, but just the right size to represent the phenomena under consideration with maximal understandability.

Educational event is divided into several sub-events, as shown in Figure 11.4, in which definitions of *I_L event* and *learning event* are also shown. *I_L event* is further divided into *simple event, reciprocal I_L event* and *influential I_L event. Preparing learning condition*, a subclass of *influential I_L event,* consists of three slots: *I event, effective L event,* and *prepared L event. I event* influences *learning action*, which causes *learning effect*, which in turn serves as a precondition for the next learning action, defined in the *required state of learning* slot of *prepared L event*.

11.4.4 Function Decomposition

We are now ready to talk about how to model the prescriptive aspects of theories. We believe most people will agree with us that building learning/instructional scenarios is a kind of design process, called Instructional Design or Learning Design. Let us see what is happening in the mechanical design community, because engineering design is the most typical form of design and has a long research history. Artifacts are composed of parts each of which contributes to the function of the whole by performing its own function collaboratively. This applies perfectly to a scenario which is composed of several sub-scenarios which are ordered in a sequence and perform their individual functions to achieve the function of the whole scenario. Bearing this in mind, let us see what has been achieved toward the development of a functional ontology.

One of the authors has been intensively involved for more than 15 years in the development of a functional ontology and its application to representing the functional structure of artifacts (Kitamura et al., 2006). The research objectives include uncovering the essential properties of a function and identifying how many kinds of functions and how many functional concepts exist out there. As a result, we have come up with about 90 concepts and a reference ontology of functions to explain the functional world. Furthermore, the idea of functional decomposition has been proposed to represent the functional structure of any artifact. There are two kinds of decomposition: in terms of what and how, and in terms of granularity. These are discussed below.

11.4.4.1 Decomposition in Terms of What and How

We decided to apply the idea of functional decomposition to our problem. One of the key techniques in the functional ontology is the first type of decomposition, in terms of *what to achieve* and *how to achieve* it. Let us take *to weld* as an example.

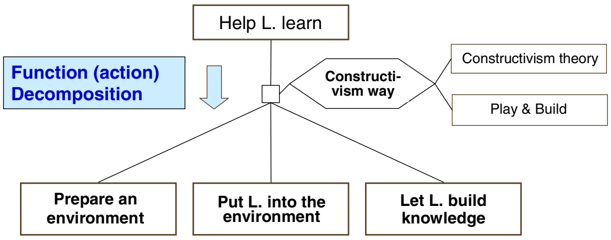


Fig. 11.5 Decomposition of *I_L event*

Although people believe *to weld* is a function, it is not correct from our theory. It is a composite concept of *to join* and *fusion way. To join* is achieved by not only *fusion way* but also by *bolt&nut way* or *glue way.* What is important here is that *to join* is a concept, independent of the way it is implemented. It even possesses high domain-independence. We call *to join* a part of *what to achieve* and *fusion way a* part of *how to achieve* it. *To cut* is not a function either. It is a composite of *to separate* and *sharp tool way*. *To wash* is another example that is not a function. It is a composite of *to clean* and *water way*. We have thus defined function as the *what to achieve* part, detaching the *how to achieve* part from the original concept. This decomposition purifies the functional concept and we end up with only about 90 functional concepts.

Conceptualization of *how to achieve* is also of value, and we introduce the notion of *way* for it. This is not a conceptualization of a sequence of sub-functions such as <*put them together, melt, cool*>, the sequence usually considered as an implementation for achieving *to weld.* The conceptualization of this sequence of actions as a "sequence of sub-functions" is not interesting at all because it is still a process. What we have done is different. We conceptualize it as a *principle* of why the sequence of sub-functions makes sense, rather than as a sequence. In terms of ontology, we conceptualize *how to achieve* as a relation between the original functional concept and the derived sequence of sub-functions, e.g., the relation between *to weld* and <*put them together, melt, cool*> in our case.

11.4.4.2 Decomposition in Terms of Granularity

Let us come back to the topic of learning with the notions of function and way in mind. *What to achieve* corresponds to *Help learners learn*, and *how to achieve* corresponds to, say, *constructivism way*, etc. Figure 11.5 shows a diagrammatic explanation of this relation. To achieve the goal by the *constructivism way*, when we read the diagram in a top-down manner, it says "I_L event of *Help learner learn* is decomposed into three sub-events: *prepare an environment, put learner into the environment* and *let learner build knowledge". .* This is the second type of decomposition in terms of granularity, which is explained in 4.6.

This implies that learning/instructional theories are understood as how to achieve, that is, a way for achieving learning goals in our conceptualization. In fact, as Figure 11.5 suggests, the same goal "Help learners learn" can be achieved by the behaviorism way, the cognitivism way, the constructivism way, etc. This is what we want to do with learning/instructional theories, that is, they are modeled in the system as alternatives to one another rather than competing and conflicting with one another. We can summarize the above discussion in Policy 5 below.

Policy 5: In the case of capturing actions and processes, detach how to achieve *from them to obtain* what to achieve*, and organize each separately. The latter should be conceptualized as "purified action" and the former as "way".*

Of course, the quality of the goal achieved is influenced by the *way* which has been applied to it. If two sheets of metal are welded, they cannot be separated, whereas if the bolt&nut way is used, they are detachable. The strength of the joint must be different from the *way* used. This would be the very point that experts are concerned with. That is, the difference would be their justification for loudly proclaiming the distinction between theories. Note, however, that we do not intend to neglect such differentiation between theories/*ways*. What we would like to do is to find a common background for theories so that users have a chance to access theories and compare them in order to choose the best one for their goals. So, we intentionally leave freedom of choice to the users, and the system would propose possible theories usable in the situation the users are in.

11.4.5 Modeling Procedures

Our preparations are now complete and we are ready to introduce our main idea of capturing procedures declaratively. The second type of decomposition in terms of granularity can be repeated until it reaches sub-functions of satisfactorily fine granularity to specify explicitly enough what should be done and in what way. Figure 11.6 depicts an example of two-way decomposition. Note that all four nodes represent *I_L events*, each of which is composed of *instructional action, learning action* and *state change*. The top node to be decomposed says "the instructor wants the learners to recognize what they are going to learn," "the learners recognize it" and the resulting state is "have recognized". There can be alternative ways to achieve the *state change* or perform the function: One is *Way1*, which explains *what to learn* and then how to learn it, and the other is *Way2*, which merely displays examples without any explanation. *Way1* is taken from Gagne and Briggs (Gagne and Briggs 1979) theory and *Way2* from Collins (Collins et al. 1989). In OMNIBUS such *ways* derived from theories are called *way-knowledge*. Because what is obtained by decomposition is also an *I_L event*, the decomposition operation can be carried out further on them. We call the resulting tree the *I_L event decomposition tree*. When you select a *way*, you have alternatives, so the links from a node to *ways* are in *"or"* relation. After the selection, on the other hand, links from the way to nodes are in "*and"* relation, as shown in Figure 11.6. Usually, authors end up with a tree whose *ways* are all determined, so picking up all the leaves (*I_L events* at the bottom) of the tree from left to right will generate an instructional/learning scenario. Furthermore, the upper structure of the tree

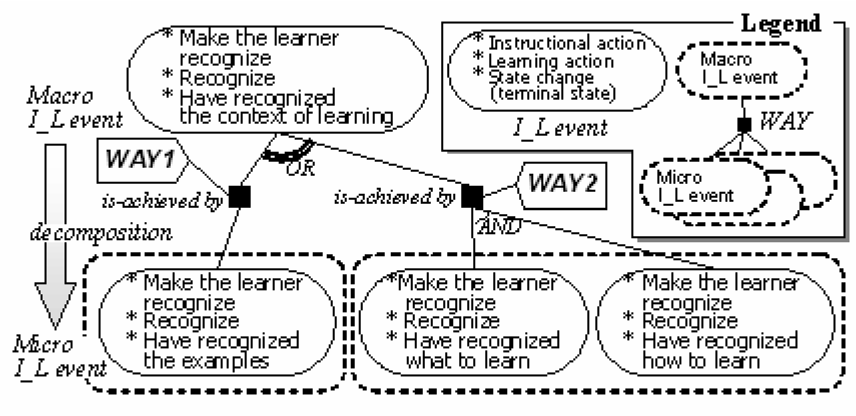

* Make the learner recognize
* Recognize
* Have recognized the context of learning
Macro I_L event
decomposition
WAY1
is-achieved by
OR
is-achieved by
WAY2
AND
Micro I_L event
* Make the learner recognize
* Recognize
* Have recognized the examples
* Make the learner recognize
* Recognize
* Have recognized what to learn
* Make the learner recognize
* Recognize
* Have recognized how to learn
Legend
* Instructional action
* Learning action
* State change (terminal state)
I_L event
Macro I_L event
WAY
Micro I_L event

Fig. 11.6 A portion of an I_L event decomposition tree

composed by intermediate nodes will give you the design rationale of the scenario, since each *I_L event* in it represents the goal of the corresponding *actions/changes*.

In functional ontology research, it is well known that accumulation of *ways*, organized in a good structure, is beneficial for understanding a design methodology. In fact, while purified functions embody domain- and implementation-independent goals, *ways* contain rich domain-specific information which is worth analyzing. Ways contain the justification of the decomposition as well as a sequence of sub-actions. In the case of the *I_L event* decomposition tree, each *way* is a compact description of instructional/learning strategy elaborated in theories. Therefore, analyzing *ways* extracted from theories contributes to understanding them from innovative perspectives. In fact, we can organize these pieces of *way-knowledge* in an *is-a* hierarchy and each of them is defined by referring to states and actions which are already defined in the ontology (Hayashi et al., 2009b). Such an organization of *ways* gives us very different perspectives from a taxonomy of theories. What is most interesting is that each of the *ways* is interpreted as a unit of strategy for instructional activity and can be directly used to build learning/instructional scenarios by decomposing the starting *I_L event*.

Imagine a situation where a lot of *ways* are stored and are used to author a learning/instructional scenario. As explained thus far, all the *ways* are described in terms of *I_L events* that are defined by referring to *states* and *actions* defined in the ontology. Therefore, the computer can easily find applicable *ways* to decompose the target *I_L event* by simple pattern-matching, and then it proposes all the candidates to the user. The user selects one he/she likes from among them and decomposes the *I_L event* to get finer-grained *I_L events*. This decomposition process is continued until executable actions are reached, to obtain a learning/instructional scenario. The scenarios thus obtained are necessarily justified by theories, and hence they should be reliable, generic and articulate. The above discussion is summarized in three policies:

Policy 6: Build action-decomposition trees using basic conceptual units obtained following Policy 4, the purified action obtained from Policy 5, and its ways.

Policy 7: Extract ways from sources of procedures, and organize them in an is-a hierarchy.

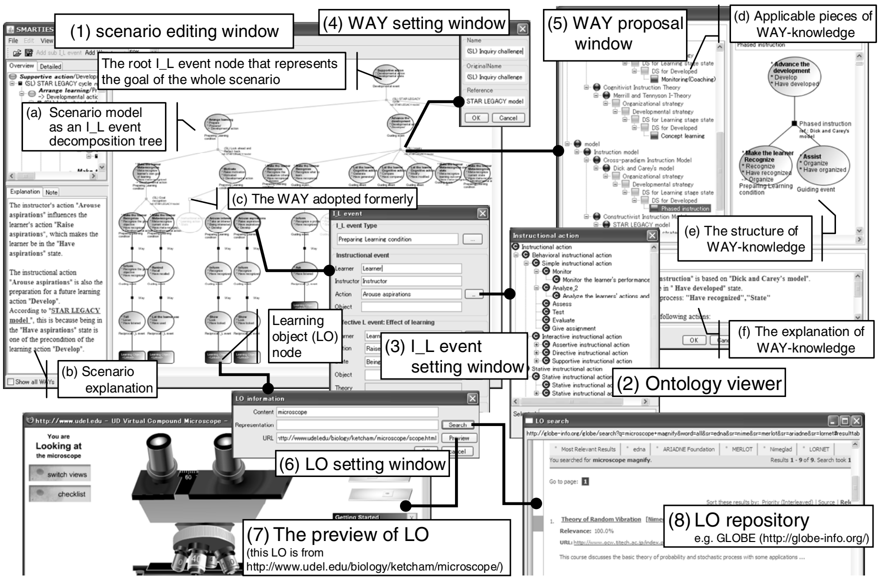
Policy 8: Constitute a problem-solving engine by considering the extension (decomposition) operation of the I_L event decomposition tree as a kind of inference.

11.4.6 From Theories to Strategies and vice versa

Ways are educational strategies that are derived from theories (theory-based), inspired by practice (empirical) or both. In OMNIBUS, a theory is specified as being a hypothesis that is supported by evidence. Learning theories are attempts to explain the learning phenomenon through a learning mechanism, and consequently through states, actions and events. A general level of mechanism is by paradigm: behaviorists see learning as an association, cognitivists as information processing, constructivists as construction through interaction, and socio-constructivists as social interaction. A more specific level is by theory: Piaget's view is construction by accommodation and assimilation, etc. Educational strategies are derived from these theories (Bourdeau et al., 2007). In OMNIBUS, each *way* has a property called *theory of reference*. This is how SMARTIES allows authors (instructional designers) to make design decisions that are explicitly linked to a theory, and to reflect on each decision they make while building a learning scenario, at the macro- or the micro-level. Several examples have been designed to illustrate the power of the OMNIBUS-SMARTIES system. The example in Figure 11.7 was inspired by Reigeluth's book *Instructional Theories in Action* (Reigeluth, 1987), which provides a variation of scenarios, based on different theories, for the same lesson in optics (concerning lenses and microscopes). In our example, our system allows a designer to select a theory on which to build a complete scenario with fine-grained learning activities, using learning objects such as a virtual microscope and a simulation in optics. The OMNIBUS-SMARTIES system welcomes all theories without any preference. The strength of a theory in education is, as in any domain, its power of explanation and prediction, and the evidence that supports it.

11.5 The Design of a Learning Activity

SMARTIES has been designed and implemented based on OMNIBUS, in line with the aforementioned policies. We will now discuss how these policies work in building SMARTIES. OMNIBUS was designed with the notion of *target-world modeling* in mind. It provides us with all the necessary conceptual building blocks to model learning/instructional activities and captures the prescriptive aspects of theories in the form of *ways,* the *way* being a unit of *I_L event* decomposition.


(1) scenario editing window
(4) WAY setting window
(5) WAY proposal window
(d) Applicable pieces of WAY-knowledge
The root I_L event node that represents the goal of the whole scenario
(a) Scenario model as an I_L event decomposition tree
(c) The WAY adopted formerly
(e) The structure of WAY-knowledge
Learning object (LO) node
(3) I_L event setting window
(f) The explanation of WAY-knowledge
(b) Scenario explanation
(2) Ontology viewer
(6) LO setting window
(8) LO repository
e.g. GLOBE (http://globe-info.org/)
(7) The preview of LO
(this LO is from http://www.udel.edu/biology/ketcham/microscope/)
Explanation
Note
Show all WAYs
You are Looking at the microscope
switch views
checklist
Make the learner Recognize
Assist
Guiding event
Preparing Learning condition
Phased instruction
OK
Cancel

The instructor's action "Arouse aspirations" influences the learner's action "Raise aspirations", which makes the learner be in the "Have aspirations" state.

The instructional action "Arouse aspirations" is also the preparation for a future learning action "Develop".
According to "STAR LEGACY model", this is because being in the "Have aspirations" state is one of the precondition of the learning action "Develop".

Fig. 11.7 Design of learning/instructional activity in SMARTIES.

Figure 11.7 shows a SMARTIES window configuration. The largest, main window (1) is the *scenario editing window*, in which authors can build a scenario model in terms of an *I_L event* decomposition tree, on the basis of the models of learning/instructional actions in OMNIBUS. SMARTIES can also help users to get leaf *I_L events* connected with LOs.

Following *Policy 1*, authors can define their own *I_L events* and *ways* in terms of *states* and *actions* defined in OMNIBUS (Fig. 11.7 (3), (4)). Then, those *I_L events* are understood by SMARTIES, which can use them in its future behavior. Although authors can define *I_L events* and *ways* in their own terms, SMARTIES cannot utilize them because it cannot understand the meaning of such terms. It can find applicable pieces of *way-knowledge* in the *way-knowledge* base, which stores *ways* defined in accordance with *policies 2* and *3*. Fig. 11.7 (5) is the *way-proposal* window, which shows applicable pieces of *way-knowledge*, sorted in order of types of paradigms, theories and strategies, for decomposing the current *I_L event*. In any situation, authors can query SMARTIES for an explanation of the object they indicate. Thanks to the proposals and explanations given by SMARTIES, authors can easily build theory-guaranteed learning/instructional scenarios as *I_L event* decompositions. Such activity constitutes the design of learning/instructional actions by SMARTIES. When LOs are connected to an *I_L event*, a learning/instructional model has been completed. *Policies 5* to *8* appear to enable SMARTIES's authoring behavior.

11.6 Concluding Remarks

We have thus far discussed the ontological approach to *target-world modeling*, taking an example of learning/instructional actions from the OMNIBUS/SMARTIES project. We summarized our experience in building the OMNIBUS ontology into eight *policies* on the basis of our long experience in ontological engineering research.

As discussed above, the difficulties we had to overcome were reconciling the gap between paradigms and modeling prescriptive aspects of theories. Let us summarize how we coped with these to build SMARTIES/OMNIBUS.

11.6.1 SMARTIES Is Not an Expert System

Viewing SMARTIES in terms of its performance, it looks like an expert system. From the viewpoint of its system architecture, however, it is not. It has no rule base, no inference engine and no heuristics. SMARTIES performs only simple operations, merely reading and writing concepts defined by OMNIBUS, and performing pattern matching of *I_L events* between one to be decomposed in a scenario model and others described as the macro events in the pieces of *way-knowledge*. The pieces of *way-knowledge* that have the macro event agreeing with the *I_L event* to be decomposed are applicable ones, that is to say, applicable learning/instructional strategies from theories.

The reason SMARTIES can provide versatile functions with such simple operation is that it has been designed in accordance with the seven requirements discussed in Section 2. Roughly speaking, OMNIBUS corresponds to the knowledge base of an expert system. However, unlike expert systems which model the problem space according to the heuristics of human experts, OMNIBUS models the problem space by *target-world modeling*, that is, the problem space is modeled as objectively as possible to represent what theories say about the target world. In other words, while expert systems have tried to capture how human experts solve particular problems, OMNIBUS/SMARTIES tries to capture the problem domain as a whole by utilizing the *target-world modeling* strategy, which matches very well with the philosophy of ontology building. Needless to say, there is no idea of ontology building in conventional expert systems, which hide the underlying reasons why their rules make sense and are applicable to the situations. On the other hand, prescriptive aspects of theories are successfully captured in the OMNIBUS ontology and hence, SMARTIES can explain its own behavior and the knowledge used in most cases.

11.6.2 Qualitative Evaluation of the Model of Learning/Instructional Theories

In building OMNIBUS, we did not attempt to model theories with total precision, which is inherently impossible because of the nature of representation in a computer. We aimed, rather, at developing a common background for theories based on *engineering approximation*. Success factors here include employing state-based representation and viewing theories from the standpoint of their use rather than of what they are. Although it is true that theories are very different from paradigm to paradigm, they are captured in OMNIBUS in terms of the learner's state, which is assumed by theories. As a result, we have confirmed that our model of learning/instructional theories has two remarkable characteristics:

(1) Classification of states roughly matches that of theories in terms of paradigm. For example, it is understood that cognitivism theories refer mainly to cognitive states, while constructivism theories refer to states related to meta-cognitive activities. Statistical analysis of the states referred to by *ways* extracted from theories clearly reflects that understanding. That is, references to cognitive states by cognitivist theories represent the highest percentage when compared either with other states referred to by cognitivist theories or with references by other theories to cognitive states. The same applies to constructivism theories and states related to meta-cognitive actions. In summary, we confirmed a high correlation between state classification and theory classification.

(2) Identification of overlapping characteristics among theories: Theories in each paradigm have different features by which they are differentiated. Nevertheless, they are not completely different, since they all are about learning/instruction. In spite of this simple fact, there is no convincing explanation for their degree of similarity to one another. What we have found in the detailed analysis of *ways* extracted from theories is the existence of a reasonable amount of overlap among the states referred to by theories in different paradigms. This should be easily inferred from the fact that the percentages of the number of states referred to by constructivism theories are 6.3, 36.7, 41.4, 4.7, 0.8, 11.2 for learning stages, cognitive process state, meta-cognitive process state, attitudinal state, developmental state and external state, respectively. For details, see Table 4 in (Hayashi et al., 2009c). A similar scattering tendency is definitely found for the cognitivism case as well. These facts suggest that theories in different paradigms are not so isolated or conflicting as we would think. They could be even blended if authors wish. At least, theories in the same paradigm have a much higher blending potential. Although blending theories is an interesting option, there is no theory justifying it. Consequently, it must be done at the author's own risk.

11.6.3 Future Work

Analyzing theories in terms of ways and states is interesting because it provides us with a new insight about theories. We have started to develop a support function and environment in SMARTIES for analyzing theories in terms of strategy, action and state. The first results are available at (Hayashi et al. 2009c).

Another thing that needs doing is integration of one-to-one/many tutoring and collaborative learning. OMNIBUS/SMARTIES collects theories essentially for one learner and no theory for collaborative learning is included in it. One of the authors has been involved in ontology building for collaborative learning and its use for group formation and authoring learning materials for collaborative learning (Isotani et al. 2009; Isotani et al. 2010). A good thing is that both ontologies are based on the *I_L event*, so that they semantically interoperate and can be integrated into a unified framework. The key issue here is that interaction in collaborative learning can be viewed as any participant learns through interaction, and that view allows us to see that the other participant who gives a stimulus to the participant who learns is playing the role of "instructor" in the broad sense, whether or not he/she intends to do so. This is the reason why *I_L event* can be the core of an ontology of collaborative learning. The first result of this integration is available at (Hayashi et al., 2009a). We believe that this challenge will help to open up a new world of learning/instructional activity modeling.

References

Bourdeau, J., Mizoguchi, R., Hayashi, Y., Psyche, V., Nkambou, R.: When the Domain of the Ontology is Education. In: Proc. of LORNET 2007, User Centered Knowledge Environments: from theory to practice, The fourth annual LORNET conference I2LOR 2007 of the LORNET Research Network (2007), `http://www.licef.ca/bourdeau`

Collins, A., Brown, J.S., Newman, S.E.: Cognitive apprenticeship: Teaching the crafts of reading, writing & mathematics. In: Resnick, L.B. (ed.) Knowing, learning, & instruction: Essays in honor of Robert Glaser. Lawrence Erlbaum Associates, Hillsdale (1989)

Gagne, R.M., Briggs, L.J.: Principles of Instructional Design, 2nd edn. Holt, Rinehart & Winston, New York (1979)

Galton, A., Mizoguchi, R.: The water falls but the waterfall does not fall: New perspectives on objects, processes and events. Journal of Applied Ontology 4(2), 71–107 (2009)

Hayashi, Y., Bourdeau, J., Mizoguchi, R.: Toward a Learning/Instruction Process Model for Facilitating Instructional Design Cycle. In: Proc. of 9th IFIP World Conference on Computers in Education (WCCE 2009), pp. 138–147 (2009a)

Hayashi, Y., Bourdeau, J., Mizoguchi, R.: Towards Better Understanding of Learning/Instructional Design Knowledge with Strategy-centered Structuring. In: Proc. of ICCE 2009, pp. 91–98 (2009b)

Hayashi, Y., Isotani, S., Bourdeau, J., Mizoguchi, R.: Using Ontological Engineering to Organize Learning/Instructional Theories and Build a Theory-Aware Authoring System. International Journal of Artificial Intelligence in Education 19(2), 211–252 (2009c)

Hozo: ontology editor, `http://www.hozo.jp/`

Isotani, S., Inaba, A., Ikeda, M., Mizoguchi, R.: An Ontology Engineering Approach to the Realization of Theory-Driven Group Formation. International Journal of Computer-Supported Collaborative Learning 4(4), 445–478 (2009)

Isotani, S., Mizoguchi, R., Inaba, A., Ikeda, M.: The foundations of a theory-aware authoring tool for CSCL design. International Journal of Computers and Education 54(4), 809–834 (2010)

Kitamura, Y., Koji, Y., Mizoguchi, R.: An Ontological Model of Device Function: Industrial Deployment and Lessons Learned. Journal of Applied Ontology 1(3-4), 237–262 (2006)

Mizoguchi, R., Bourdeau, J.: Using Ontological Engineering to Overcome Common AI-ED Problems. International Journal of Artificial Intelligence in Education 11(2), 107–121 (2000)

Mizoguchi, R., Hayashi, Y., Bourdeau, J.: Inside Theory-Aware & Standards-Compliant Authoring System. In: Proc. of SWEL 2007, pp. 1–18 (2007)

Mizoguchi, R.: Yet Another Top-level Ontology YATO. In: Proc. of the Second Interdisciplinary Ontology Meeting, pp. 91–101 (2009)

Murray, T., Blessing, S., Ainsworth, S.: Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive & Intelligent Educational Software. Springer, Heidelberg (2003)

Reigeluth, C.: Instructional Design Theories in Action. Erlbaum Associates, Hillsdale (1987)