

# Confronting Some Ontology-Building Problems in Educational Topic Map Authoring

Darina Dicheva and Christo Dichev

Winston-Salem State University, 3206 E J Jones Computer Science Building  
Winston Salem, NC 27110, USA  
{dichevad, dichevc}@wssu.edu  
<http://compsci.wssu.edu/iis/>

**Abstract.** Ontologies are a key technology for achieving personalization in the educational hypermedia. In this paper we discuss some common problems the authors of Topic Map-based learning content experience in building subject ontologies, i.e. in structuring and articulating the conceptual model of the intended online learning material. The discussed problems are not specific to educational topic map authoring; they are related to the creation of educational ontologies in general. We discuss the re-design of the educational topic map editor TM4L, focusing on the provided authoring support for ontology building.

## 1 Introduction

Ontologies are a key technology that can facilitate Web information processing by supporting semantic structuring, annotation, indexing, and search. Ontologies allow organization of learning material around semantically annotated topics, which enables semantic querying combined with intuitive navigation and access to the learning resources. They provide the necessary grounds for achieving better personalization in the educational hypermedia and Web-based applications by enabling more adequate and accurate representations of learning material and contexts of its use, from one side, and of learners and their goals, from another, as well as more efficient access and navigation through the learning resources. We have developed an authoring tool, the TM4L Editor [3], which enables the development of ontology-aware courseware based on topic maps [6].

In the Topic Map (TM) paradigm an ontology is an accurate description of the essential entities and relations which are found in the modeled area, and can be represented as a set of *topics* linked by *associations*. Therefore, the TM technology is well suited for structuring learning material around subject ontologies. Educational Topic Maps (ETM) can use subject domain ontologies for classification of learning content and resources. This classification involves linking the content to relevant ontology terms (concepts), i.e. using the ontological structure to index the content.

We have been experimenting with the TM4L Editor by creating educational topic maps for various university courses. In this paper we discuss our main findings related to educational ontology building and how we have used them to re-design TM4L. We

start with a brief description of topic map authoring with TM4L. Then we discuss the main problems authors face when creating ontology-aware, TM-based courseware and the grounds for these problems. Finally, we propose strategies for overcoming some of the authoring problems and describe their use for re-designing TM4L.

## 2 Creating Ontology-based Learning Content with TM4L

### 2.1 The TM4L Editor

The Topic Maps for e-Learning (TM4L) environment contains a Topic Map Editor that allows users to build ontology-driven learning repositories based on the Topic Maps model. It provides ontology and metadata engineering capabilities coupled with basic document management facilities [3]. The learning content created by the Editor is compliant with the XML Topic Maps (XTM) standard [8] and thus interchangeable and interoperable with any standard XTM tools. We aimed at providing an environment that can support a topic map lifetime, from the design and creation through maintenance and onto evolution. At present TM4L facilitates the building of educational TMs by providing support for creating and updating hierarchies of topics, topic types and instances, relationships between topics, and scopes, as well as for relating learning resources to topics.

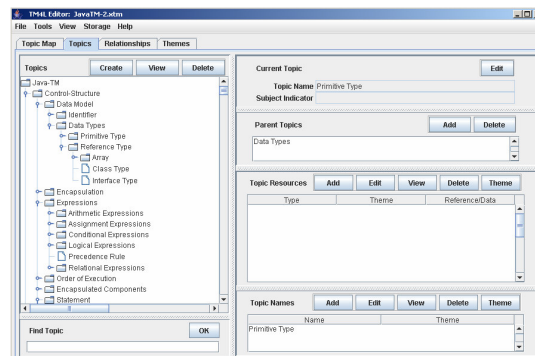


Figure 1. A screenshot from the TM4L Editor interface.

The heart of an educational topic map is a subject ontology (in the considered domain). As most existing ontology editors [1] (e.g. Protégé <http://protege.stanford.edu/>), TM4L provides a graphical user interface that displays the learning domain conceptual structure as a hierarchy of topics (see Fig. 1). Using the interface, the author can manipulate (add, delete, and modify) concept/topics, resources, and relations or perform other TM-related operations such as merging, browsing, searching, and scoping. TM4L GUI uses the *tab* metaphor where each tab is associated with a different view on the topic map: Topics view, Relationships view, and Themes view. The Topics view provides a topics tree for editing topics in context, while the Relationships view

provides a three-pane interface for editing relationship types, roles and instances. The purpose of themes is to allow the topic map author to express the limits within which the topic characteristics have validity. To assist in defining themes, the Theme view offers a two-pane interface for viewing the effect of the currently set themes by hiding/displaying properties within the selected theme. TM4L is free software that can be downloaded from <http://compsci.wssu.edu/iis/nsdl/>. TM4L GUI is currently translated in 8 languages.

## 2.2 The Study

As part of the TM4L Editor evaluation, we conducted a study to find out what are the major difficulties that authors of educational topic maps face. ETM authors are typically instructors who create the maps for the courses they teach. We invited a number of instructors to create learning content using the TM4L Editor. Nine topic maps were created in the study for the following course units:

1. *Groups* (Modern Algebra)
2. *Basic Counting Principles* (Discrete Mathematics)
3. *Number Systems* (Software Systems)
4. *Introduction to AI* (AI)
5. *Integrated Circuits* (Computer Hardware)
6. *XHTML* (Internet Technology)
7. *Storage Management* (Operating Systems)
8. *Prolog* (AI)
9. *Introduction to Java* (Computer Programming I)

The goal of the study was twofold: from one side to evaluate the TM4L Editor's interface and functionality, and from another – to provide feedback about the difficulties authors encounter in TM creation. In the study we found that the authors, being instructors and competent in the subject area, didn't have problems in selecting appropriate resources. The major difficulties that they faced were related to ontology building, i.e. to content conceptualization and classification, and to identifying and naming relationships between topics. Indeed, since there is no rule for determining the "best" criterion for classifying topics, authors had to make necessarily subjective decisions on how classes can be broken down into subclasses. We discuss these difficulties in the next section.

## 3 Difficulties in Building Educational TMs

### 3.1 Difficulties in Modeling the Learning Content Structure

In a typical courseware organization, learning content is laid out in a tree-like structure of course units (course, modules, lectures, etc.) much like the structure of the Windows file system. This approach has been adopted from the traditional textbooks organization (chapters, subchapters, sections, etc.). Usually the course units are linked with some relationships based on the author's notion of classification. *It is appealing*

to think that the course units can be organized in a taxonomy based on a hierarchy of domain concepts. However, this is not true in general. Frequently the concept structure used to organize the learning content is not a proper hierarchy, and the concepts naming the sub-units of the learning material do not represent more specialized content of their “parents” [4]. This is well illustrated in Table 1, which presents fragments of the contents of four different well known textbooks on Prolog. In the first case the topic “Lists” is under the topic “Recursive programming”; in the second, it is under the topic “Lists, Operators, Arithmetic”; in the third case it can be found under “Using Data Structures”, while in the last case, it is under “Unification, Recursion and Lists”.

**Table 1. Fragments of Prolog textbook contents.**

<p><b>Sterling and Shapiro</b></p> <p>3. Recursive programming</p> <p>3.1 Arithmetic</p> <p>3.2 Lists</p> <p>3.3 Composing recursive programs</p> <p>3.4 Binary trees</p> <p>3.5 Manipulating symbolic expressions</p>	<p><b>Ivan Bratko</b></p> <p>3. Lists, Operators, Arithmetic</p> <p>3.1 Representation of lists</p> <p>3.2 Some operations on lists</p> <p>3.3. Operator notation</p> <p>3.4 Arithmetic</p>
<p><b>Clocksinn and Mellish</b></p> <p>3. Using Data Structures</p> <p>3.1 Structures and Trees</p> <p>3.2 Lists</p> <p>3.3 Recursive Search</p> <p>3.4 Mapping</p> <p>3.5 Recursive Comparison</p> <p>3.6 Joining Structures Together</p> <p>3.7 Accumulators</p> <p>3.8 Difference Structures</p>	<p><b>Paul Brna</b></p> <p>5. Unification, Recursion and Lists</p> <p>5. 1 Unification</p> <p>5.2 Recursion</p> <p>5.3 Lists</p> <p>5.3.1 How to construct/deconstruct a list</p> <p>5.3.1.1 List Destruction:</p> <p>5.3.1.2 List Construction:</p> <p>5.3.1.3 Bigger Chunks:</p> <p>5.3.2 The Empty List</p> <p>5.3.3 Some Possible Matches</p> <p>5.3.4 A Recursive Program Using Lists</p>

Obviously, the topic “Lists” is not a kind of “Recursive programming”, nor a kind of “Using Data Structures”. On the other hand, the topic “Lists” could be considered a subclass of “Structures and Trees”, and likewise “The Empty List” is an instance of “Lists”. Strictly speaking, “Lists” is not a proper sub-class of any of the parent topics in the examples listed above; it is rather a subclass of the topic “Data Structures”.

These examples confirm that the order in learning content classifications is often subjective and arbitrary and could be easily altered, which in turn means that it doesn’t represent a true hierarchy. Apparently the subjectivism results in ad hoc concepts organizations. Moreover, authors’ intuition about where to place a unit sometimes is inconsistent with a broadly adopted structure. This may result in putting a unit in an unexpected for the learners place. Additional complication is that both the organization and the names of the concepts are subject of change over time. An example is the ACM Computing Classification System ([www.acm.org/class/1998/overview.html](http://www.acm.org/class/1998/overview.html)), which has been changed several times since its first publication. This makes impractical any attempts to derive a durable structure of instructional units based on the *folder-sub-folder* pattern. A more permanent and predictable approach is to capture

the domain concepts and their relations based on the domain ontology rather than on ad hoc collection of *folders*.

### 3.2 Difficulties in Topics Identification

The structure of the learning content reflects the author's concept of systematization. The topics and their relations make sense to the author in the context of their thoughts and goals at the time of creation. Being subjective, they are with uncontrolled terms and of variable quality. There are no rules to constrain the authors to use specific names for describing the content. Therefore one challenge is to select an appropriate name for each group of resources. For instance, looking at the tables of content of some online Prolog resources such as [http://en.wikibooks.org/wiki/ Programming: Prolog](http://en.wikibooks.org/wiki/Programming:Prolog) and [www.amzi.com/AdventureInProlog/index.htm](http://www.amzi.com/AdventureInProlog/index.htm), you can find in one titles "Recursion" and "Arithmetic", and in another "Recursive rules" and "Math, Functions and Equality". This reflects some known principal problems with identification and naming of concepts/topics. The naming problems cause in turn problems related to the search for resources. An example is the use of different titles to represent the same topic. In such cases, a search for the term "Pattern Matching" will not pick up resources indexed with the term "Unification".

### 3.3 Difficulties in Articulating Relations

As we mentioned the authors had most difficulty in deciding what type of relationships to use in constructing their TM and how to name them. Authors typically follow the structure of the content in the textbook they use. Typical learning content structures are incoherent combinations of taxonomies, partonomies, and other unnamed schemes. The majority of available electronic versions of learning material continue this practice of the pre-Web era. Table 2 summarizes the number of topics and of association types (together with their instances) for the nine topic maps created in our study.

Some TM authors pointed out that in many cases they had found it difficult to determine the exact type of a particular "more inclusive-less inclusive" kind of relation they are dealing with. In such a case they place the new topic as a child of an already created topic in the displayed topic hierarchy. The extreme case of this was demonstrated in the topic maps for "Groups" (Unit 1) and "Basic Counting Principles" (Unit 2), where beside the "created by" relation, all other relations between the defined topics were defined as instances of the default hierarchical type (the one used in the topic tree construction): 74 in the first topic map and 45 in the second.

The parent-child classification reflects the title-subtitle (section-subsection) tradition established by the conventional textbook organization and the created TM-based learning material mirrors the context in which authors used to see the composing items. The problem is that an organized collection of learning resources representing contextually related topics is often difficult to translate into a conventional hierarchical structure. When the relationship between two topics is not hierarchical, the problem is even harder. Apart from articulating it, the authors have to decide how to name it. The difficulty comes from the fact that differently from concepts, which are gener-

ally named by terms from the considered subject area, there are no established and agreed names for relations. Usually they are named by connotative words and there is a range of words that can express their meaning. The open choice of names can lead to problems for the reusability and interoperability of the created courseware.

**Table 2. Relation types in the seven Topic Maps.**

Unit #	Topics	Relation Types (# relations)	Unit #	Topics	Relation Types (# relations)
1	90	“created by” (12) “more general-less general” (74)	2	54	“created by” (5) “more general-less general” (45)
3	93	“derived from” (1) “described by” (1) “part of” (38) “represented by” (2) “simplifies” (1) “instance of” (24)	4	41	“described by” (1) “extended to” (1) “implemented by” (2) “processed by” (2) “superclass-subclass” (10) “instance of” (18)
5	67	“composed of” (3) “converted to binary by” (4) “converted to decimal by” (1) “interpretation in decimal” (4) “operates on” (10) “representation” (4) “superclass -subclass” (47) “instance of” (50)	6	113	“based on” (1) “combinations of” (3) “instance of” (33) “superclass-subclass” (8) “invented by” (2) “related to” (3) “same as” (15) “used for” (3)
7	387	“instance of” (67) “superclass-subclass” (19) “related to” (7) “same as” (15) “similar to” (3)	8	51	“based on” (1) “created by” (13) “part of” (6) “instance of” (52)
9	79	“superclass-subclass” (23) “based on” (7)			

### 3.4 The ‘Default Relation’ Syndrome

Another problem related to ontology authoring comes from the fact that in most ontology editors only one hierarchical relation type (typically “class-subclass”) is displayed as a tree. As a result, only the topics linked with relationships of that *default* type would be displayed in the constructed tree. Therefore the authors cannot see all created topics arranged in a nice tree-like structure. An attempt to “fix” this problem brought the author of the “Number Systems” (Unit 3) topic map to define relationships of two types between some topics instead of one – “instance-of”, which was the proper one and the false “class-subclass”, introduced only for the purpose of displaying all the topics in a hierarchy. The user was using the Editor’s visualization capability for one (the default) relation to express the hierarchical structure of another.

Another reason for defining most of the hierarchical relationships in the created topic maps as “superclass-subclass” was the ease of attaching children topics in the (default “superclass-subclass”) topic map tree as opposed to explicitly creating and naming relationships in the separate “Relationships” panel.

## 4. Redesigning the TM4L Editor to Support Ontology Building

The input from the conducted study suggested directions for improving the TM4L Editor. The difficulties in articulating and naming relations prompted us to include predefined relationship types in TM4L. For that purpose we developed an empirically justified minimal ontology for TM based e-Learning content. Further on, the ‘default relation’ syndrome suggested the replacement of the single topic hierarchy displayed in the Editor with three different ways of visualizing a topic map based on the three most typical relation types – “superclass-subclass”, “part-whole”, and “instance of”.

### 4.1 The Predefined Minimal Relation Set

To assist better the authors of educational topic maps, we introduced a predefined set of relations. This set is intentionally small however it can be extended with an arbitrary number of associations. Currently it consists of five relations: three hierarchical and two ‘horizontal’ relations. The three hierarchical relations are the most commonly used semantic relations “superclass-subclass”, “whole-part” and “class-instance”. The other two relations, “related to” and “similar to”, have been proposed for capturing concept structures typical for e-learning content [4]. Note that we are not attempting to define an ontology covering the complex domain of e-learning, including the creation of learning designs, etc. Such efforts and their results can be found, for example in [5] and [7], among others. Our focus is on defining a minimal set of generic relations expressive enough to allow creating ontologies for structuring learning content.

What relations hold between certain pairs of concepts is sometimes unclear for practitioners in the area. For example, Prolog is related to “artificial intelligence,” “logic” and “constraint programming.” But how exactly? To avoid introducing ad hoc relations with arbitrary names, we provide the “related to” relation. It captures associative connections between concepts, such as “refers to”, “is relevant to”, etc. Similar to the W3C SKOS scheme (<http://www.w3.org/2004/02/skos/>) this relation carries weak semantics. Its purpose is to express the fact that two concepts are related in some way, and that the relationship should not be used for creating a hierarchy. The relation “similar to” is intended to capture in a generic sense semantic similarity between concepts that have a close meaning or express different degree of a certain state or quality (e.g. *advanced* and *complex*). The metaphor ‘horizontal’ is used to suggest that the relations allow creating links between branches in a hierarchy of concepts.

The basic intuition is that the five relations “superclass-subclass”, “whole-part”, “class-instance”, “related-to” and “similar-to” represent a sufficient basis of generic relations for creating ontologies for educational topic maps. This way we enable authors to apply predefined relations between the entities, relieving them from the need to create dummy or incorrect relation types.

### 4.2 The 3-Perspective Concept Tree View

Instead of adopting a single “perspective” on concept grouping, the re-designed TM4L GUI supports three basic hierarchical views. This way, the authors are able to create various classifications of certain concepts. For example, “operators” can be classified by *arity* (unary, binary and so on) or by type (arithmetic operator, Boolean

operator, String operator, and so on). By enabling different perspectives, we can model different classifications of topics.

The concept/topic tree in the Editor's Topic Panel can be seen in three different views - *Taxonomy*, *Partonomy* and *Topic Typing*. In each of these views relationships of the corresponding *default* type (i.e. "superclass-subclass", "whole-part", and "class-instance") are automatically created/deleted during the process of topic tree editing.

### **4.3 Merging Topic Maps**

Ontology merging is a key concept in ontology engineering that is explicitly built into the TM standard. TM4L merging allows a modular approach to TM creation. All of the created concepts, relationships and resources will still be present; the merging process neither changes nor eliminates them. The key difference is that the original conceptual structure has been enriched consistently with new concepts and relations.

## **5. Authors Support in TM4L**

### **5.1 Support for ontological classifications and perspectives**

It is tempting to think that the classification schemes utilized by the libraries can be adapted in a simple way for use in e-learning. However, in e-learning environment the classification structure should account for multiple perspectives depending on the context in which the information is being sought. An implied goal in re-designing TM4L was to enable the representation of different *perspectives* of a domain.

The lack of a shared understanding and consistency in using concepts on a textbook and course level might be compensated by using objective domain conceptual structures (domain ontologies). From this viewpoint we conceive a domain ontology as a *conceptual reference system*, with a collection of concepts, relations between concepts and classification hierarchies. The resulting conceptual schema could serve as an aid for integrating related concepts (terms) from different repositories. TM4L provides support for authors that want to create ontological classification. Differently from typical ontology editors, our model includes three basic concept hierarchies: "superclass-subclass", "whole-part" and "class-instance". In this way authors are able to create conceptual structures that include various classifications of certain concepts. Authors can create typed concepts (with variant names) and an arbitrary number of resources linked to them. They can add any number of relations including transitive and symmetric relations in conjunction with the predefined classification hierarchies.

### **5.2. Reusing Ontologies**

One way to minimize the cost of ontology-based learning content is to reuse already created ontologies. The domain ontology component whose development is costly is *more stable* and therefore *reusable*. It is stable because classification based on a domain ontology is objective. Our framework supports reuse not only of learning resources but also of domain knowledge and instructional knowledge.

We can view the domain conceptual structures as independent information resources in their own right. An existing structure can then be overlaid on different information pools (using the TM4L editing functionality), or merged with other TM (using its merge functionality) capturing different viewpoints on the same collection. An existing structure can be a source for new perspectives or used for interchange.

TM4L supports ontology reuse through merging. In addition to promoting common understanding of concepts and avoiding redefinition of terms, it allows adding new meaning to preexisting course structures. For example, an existing conceptual structure implemented as a TM can be integrated with a course centered topic map by merging them with additional mapping when necessary. We used this approach to create the Java topic map, driving the Java Portal (<http://iiscs.wssu.edu/p4j/>): by merging the separately created Java ontology and the course topic map, developed specifically for the CSC1311 Programming 1 course and using Gaddis' book 'Starting Out with Java'.

### 5.3 “One Ontology-Many Containers” Educational TM Model

The conceptual representation of the domain is a key factor in our framework, which is built on the assumption that a rational structure will facilitate a meaningful representation and use of learning resources as well as the process of searching them. *We argue that a single standard classification of learning materials in a subject area is a weak strategy from a practical perspective.* In general, a classification is simply one particular rationalization of the relationships in a given domain that satisfies a group of individuals. Any classification, no matter how it is constructed, represents one possible view on a domain and thus one potential way to organize a learning collection. Any field of knowledge may be classified from different viewpoints or contexts resulting into different classifications. For each particular task the authors choose to represent one particular view of the domain, therefore, a classification is aimed - in one way or another - to support a given viewpoint at the expense of other views.

We can think of two basic perspectives on a course topic map: a *structure of concepts* describing the subject area and a *collection of text units* (chapters, subchapters, sections) storing information about the concepts in a specific format. (Note that *it is a common mistake to confuse a container for the concept(s) contained.*) It would be difficult to argue that only one of these classifications is an accurate representation of the domain and the other is not. The selection of categories in a classification depends on the viewpoint/context of the community of individuals in which the classification is created and used. In the e-learning domain, a classification based on a module/lesson structure is not less significant than its conceptual counterpart.

This approach however poses a new problem. By extending the domain ontology conceptual structure with a specific course structure the resulting topic map turns to be course specific. This of course is undesirable with regard to the reusability of the topic map. Hence we chose to create independently two topic maps – an ontology topic map and a course structure topic map. This is the basis of our proposed ‘One Ontology – Many Containers’ model of educational topic maps – the separation of the content structure into “public” and “private”.

There are two major benefits of separating the initial representations of the two classifications. The first one is that the ontology TM that represents the conceptual

structure of the learning content can be built and used independently of the structure of a specific course. The second benefit is that the ontology TM can be merged afterwards with various TM representing different course structures.

## 6. Conclusion

In this paper we outline the evolution of the TM4L Editor based on some practical observations. The focus is on the difficulties authors face when creating ontology-based courseware with TM4L, on the lessons learned and on the proposed strategies for overcoming some of them. Many of our findings may pertain to other ontology-based educational environments.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DUE-0333069 “NSDL: Towards Reusable and Shareable Courseware: Topic Maps-based Digital Libraries” and Grant No. DUE-0442702 “CCLI-EMD: Topic Maps-based Courseware to Support Undergraduate Computer Science Courses.”

## References

- [1] Denny, M. (2002) *Ontology Building: A Survey of Editing Tools*, O'Reilly XML.COM, Retrieved online 02/10/2006 at <http://www.xml.com/pub/a/2002/11/06/ontologies.html>
- [2] Dichev C. & Dicheva D. (2005). Contexts as Abstraction of Grouping, *Workshop on Contexts and Ontologies, 12th National Conference on Artificial Intelligence, AAAI 2005*, July 9-13, 2005, Pittsburgh, 49-56.
- [3] Dicheva, D. & Dichev, C. (2006) TM4L: Creating and Browsing Educational Topic Maps, *British Journal of Educational Technology*, 37(3), 391-404.
- [4] Dicheva D. & Dichev C. (2005). Authoring Educational Topic Maps: Can We Make It Easier? *5th IEEE International Conference on Advanced Learning Technologies, ICALT 2005*, July 5-8, 2005, Kaohsiung, Taiwan, 216-219.
- [5] Grandbastien, M., Gavignet, E. (1994). ECSA.- An environment to design and instanciate learning material, In: *Design and Production of Multimedia and Simulation Based Training Material*, Kluwer Academic Publisher, Amsterdam, 31-44.
- [6] Park, J. & Hunting, S. (2002). *XML Topic Maps: Creating and Using Topic Maps for the Web*, Addison-Wesley.
- [7] Paquette, G. (2003) *Instructional Engineering for Networked-based Learning*, Pfeiffer Wiley Pub.
- [8] XTM. ISO/IEC 13250:2000 *Topic Maps: Information Technology*, Retrieved online 02/10/2005 at: [www.y12.doe.gov/sgml/sc34/document/0323.htm](http://www.y12.doe.gov/sgml/sc34/document/0323.htm).