

# Knowledge Domains: A Global Structuring Mechanism for Learning Resources in WBT Systems

Denis Helic, Hermann Maurer and Nick Scerbakov  
Institute for Information Processing and Computer  
Supported New Media  
University of Technology Graz  
Infeldgasse 16c, 8010 Graz, Austria

**Abstract:** All the data structures previously introduced by different WBT systems were "training oriented" so to speak. Those data structures can be perceived as different training paradigms reflecting mainly different ways of accessing and working through a subset of learning material by learners. Primitively, a "training oriented" data structure mainly prescribes reading the document "A" before the document "B" and does not reflect a possible situation that "A" may be the documentation of the software module developed in the scope of a software project "B". Nevertheless, users often need a general overview and access to all learning resources provided by a particular WBT system regardless on any training data structure where those learning resources are reused within. The WBT-Master, a novel WBT system, supports such global structure of learning resources in the form of so-called Knowledge Domains.

## 1. Introduction

From the functional point of view, the term "WBT" refers to systems [Andrews et al., 1995] [Helic et al., 2000] that support at least:

- authoring and publishing of high quality multi-media courseware
- repository of learning resources available world-wide and anytime by means of WWW [Berners-Lee et al., 1994]
- repository containing personal info on peer helpers, on-the-fly contributions of previous users, discussions on interesting topics, etc.
- support for personal definition of learning goals and variety of learning methods
- synchronous and asynchronous communication and collaboration among learners and between learners and expert
- database of previously answered questions to automatically answer similar ones
- testing of knowledge level
- customization of courseware to the knowledge level and the cognitive style of the learner
- IPR protection and brokerage of relevant learning services
- progress tracking and certification as support for assessment and recognition aspects of the acquired skills.

Technically, a WBT system is a system which utilizes a large repository of multi-media ( preferably HTML ) documents, where information is structured according to a particular hypermedia data model[Cailliau, 1995] [Fernandez et al., 1997] [Helic et al., 2000] . The first WBT systems structured information according to the most simple node-link data model utilized by the WWW. Recently, new WBT systems were introduced utilizing more powerful data modeling facilities in order to overcome well-known problems of the basic node-link data model [Maurer et al., 1996] . WBT-Master [Helic et al., 2000] is one of those systems. The WBT-Master utilizes the unified HM-Data Model[Helic et al., 1999]and data structures supported by this model. Data structures, such as Learning Unit, Learning Course, Learning Goal, Discussion Forum [Helic et al., 2000] , etc., are supported by this model. Such data structures are well-

defined chunks of hypermedia information that may be reused in different contexts. Moreover, they provide the satisfactory authoring interface and impose a convenient navigational structure on the top of hypermedia information encapsulated in it. All other information services in WBT systems such as information searching, synchronous and asynchronous communication, collaboration, etc. are implemented as "add on" software packages using often different protocols ( Server Site Scripts, Client Site Scripts, Java Applets and Applications, etc ) [Helic et al., 2000] .

From the user point of view, a WBT system is a system which supports different user roles according to basic functionality which is required. Most common definition of user roles in WBT systems may be seen as follows [Helic et al., 2000] :

- Learners utilize WBT information services in order to improve their knowledge and skills. They are motivated to use the system services by a particular Learning Goal in mind. In a simplest case, the Learning Goal may be just a wish to pass an examination and to get some credits.
- Tutors manage the learning process on some particular subject. Tutors select an appropriate learning strategy (say, Learning By-Doing, Situation-Oriented Learning, etc) to achieve better learning results. Tutor control learners' progress with the material, offer additional materials and point learners to fellow helpers if necessary, examine learners' knowledge and acquired skills, etc.
- Authors contribute to courseware repository creating and publishing documents, combining the documents into navigable structures (courseware libraries, courses, learning modules, etc.).

Thus, one of the basic scenarios in WBT systems can be seen as follows:

- imposing different types of data structures on top of existing collections of Learning Resources, which is done by authors, or looking from the other point of view, as reusing of Learning Resources in different contexts
- customizing of Learning Resources by tutors to support reuse of Learning Resources in different contexts, as well as to adjust Learning Resources to different learners' preferences and needs
- working through Learning Resources to achieve a particular Learning Goal which is done by learners.

However, all the previously discussed data structures were "training oriented" so to speak. Thus, the previously mentioned Learning Unit, Learning Course, Learning Goal, Knowledge Card, etc., can be perceived as different training paradigms reflecting mainly different ways of accessing and working through a subset of learning material by learners. Primitively speaking, a "training oriented" data structure consisting of documents "B" and "C" mainly prescribes reading "B" before reading "C" and has nothing to do with a possible situation that "C" may be documentation on a software module implemented by the programmer "A" for a project "B". Often, users need a general overview and access to all learning resources provided by a particular server.

Let us just discuss the following situation. Suppose a software organization maintains a big repository of software technical documents. Obviously, elements of the repository are valuable Learning Resources and may be reused for training as such or as components of other Learning Resources ( say, as members of Learning Units, resources for Learning Goals, contribution to discussion forums, etc. ). Moreover, information on people maintaining documents and source codes, is also a very valuable Learning Resource which can be reused as references to peer helpers having experience in the subject area. At the same time, localization of a particular Learning Resource ( providing it was not previously reused for training ) constitutes a rather difficult problem which can be solved only by structuring the repository on meta-level invariantly to any training paradigm. Using the same primitive language as above, we can say that the knowledge: "C" is a technical description of the software module implemented by the programmer "A" for the project "B"; should be kept independently of reusing "C", "B" and "A" for training purposes.

The WBT-Master supports such global structure of learning resources in the form of so-called Knowledge Domains. The rest of this paper describes the main ideas behind the concept of Knowledge Domains, as well as some of the implementation issues.

## **2. Knowledge Domains**

Generally, a knowledge domain is a special hypermedia structuring paradigm which is based on the concept of separating structure and content.

Each Knowledge Domain is a set of documents belonging to a number of predefined semantic categories. For the previously discussed example, we could speak about three semantic categories: "Author", "Module" and "Project". We can also say that a document "C" is an instance of the category "Module", document "B" is an instance of the category "Project" and the document "A" is an instance of the category "Author". Speaking in general terms, we can say that each semantic category is linked to a set of learning resources which are called instances of the category.

Further, each Knowledge Domain explicitly defines a number of so-called semantic relationships between the semantic categories contained in this Knowledge Domain. Again, for the previously discussed example we could speak about the following semantic relationships: "Author Modules" is a relationship between the "Author" and "Modules" semantic categories, which denotes that a particular "Author" implemented corresponding "Modules"; "Project Modules" relationship relating the "Project" and the "Module" semantic categories and denoting that a particular "Module" is a part of a certain "Project". Applying the concept of semantic relationships to instances of semantic categories means interrelating these instances according to relationships between their corresponding categories. Now, we may say that the document "C" is related to the document "A" by means of the "Author Modules" semantic relationship. Similarly, the document "C" is related to the document "B" by the means of the "Project Modules" semantic relationship.

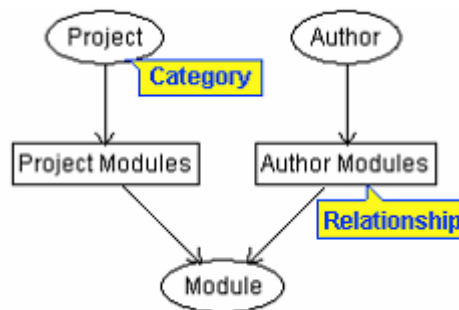


Figure 1: Knowledge Domain Schema

Thus, each particular knowledge domain is a collection of WBT-Master training resources ( documents, learning units, individual users, etc. ) which are structured using a predefined template called the knowledge domain schema.

A knowledge domain schema may be seen as a definition of all categories and all possible semantic relationships between them.

Learners may browse and search Learning Resources, reused and interrelated with other Learning Resources, as instances of semantic categories.

Thus, we may see that the concept of Knowledge Domain supports three different aspects of working with systems:

- defining a Knowledge Domain Schema, which includes defining a number of semantic categories and a number of semantic relationships between the previously defined semantic categories
- reusing resources as instances of different semantic categories and interrelating them with other learning resources according to a previously defined Knowledge Domain Schema
- browsing and searching of learning resources reused as instances of previously defined semantic categories.

### 3. Defining a Knowledge Domain Schema

A Knowledge Domain Schema defines a data structure in the form of so-called semantic categories and semantic relationships between these categories. Any training resource added to the knowledge domain is perceived as an instance of one particular category and, thus, inherits all properties defined for the category.

The schema provides a number of attributes ( data items ) which should be provided for any instance of a particular category and types of relationships which should be installed between the new instance and existing instances of other categories.

Definition of a Semantic Category includes definition of a number of attributes, which are properties of instances of the Semantic Category. An attribute is a standard key-value pair. A value of an attribute is defined to be of a specified type, i.e., a value may be a string, a number or a selection from a list of possible values. For example, the "Author" may have just one associated attribute: Name ( String ). Similarly, the category "Module" may have two associated attributes - Programming Language ( selection from a list of languages ) and Name ( String ). Looking from another point of view, we can say that each instance of the category "Author" is provided with attribute "Name".

Definition of a Semantic Relationships includes the selection of two categories which participate in the relationship. Each semantic relationship defines 1:n relationship between instances of two different categories. A category participating with a single instance in each relationship is called "Owner" of the relationship. A category participating with multiple instances in each relationship is called "Member" of the relationship. Each instance of a Member is obligatory related to a certain instance of Owner, this reference is installed whenever a new Member is created. For instance, in the example above, the relationship "Project Modules" may relate a single instance of "Project" to an arbitrary number of "Modules", or looking from another perspective, each instance of "Module" is obligatory related to a certain "Project". This relationship will be created whenever a new "Module" instance is put into the Knowledge Domain. Similarly, the relationship "Author Modules" relates a single instance of "Author" with an arbitrary number of instances of "Module".

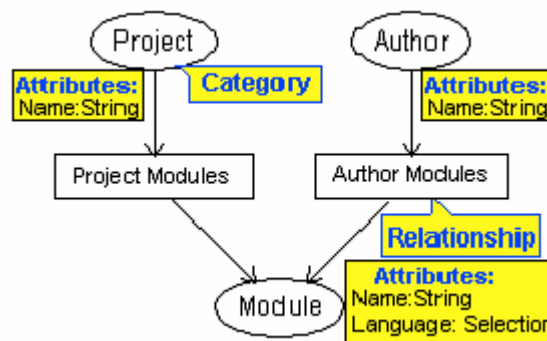


Figure 2: Categories and Relationships in a Knowledge Domain Schema

Existing of knowledge domain schema facilitates two rather important processes: structuring of a knowledge domain and browsing a knowledge domain. Whenever a new instance of a category is created, the system may assist the user with offering special forms for providing attribute values and for selecting other instances to be related to the new one. The system may also check whether all necessary information is provided and prompt the user if necessary.

From the browsing point of view, existence of a domain schema provides very useful information on a current document position within the knowledge domain and on semantic of links emanating from or pointing to the document.

#### 4. Instancing Semantic Categories

The knowledge domain schema defines common properties of all the category instances. Any resource may be inserted ( stored ) into a particular knowledge domain as an instance of predefined category. Thus, a responsible author simply selects an existing knowledge domain and a predefined category for a new resource and the system guides the author through the process of defining attributes and necessary relationships. For example, if a new instance of the category "Module" is created, the system automatically request to select a programming language ( attribute predefined for the category ), and to provide references to the module author and a certain project ( relationships predefined for the category ). This, of course, facilitate creating of well-structured repositories.

## 5. Browsing Instances of Semantic Categories

The concept of well-structured Knowledge Domains facilitates also browsing and searching the resources reused as instances of semantic categories. Thus, for example, whenever a user access the document "Module 01/01", the system automatically provides:

- information on attributes attached to this document,
- references to instances of other semantic categories which are related to this one,
- next/prior navigational tools, etc.

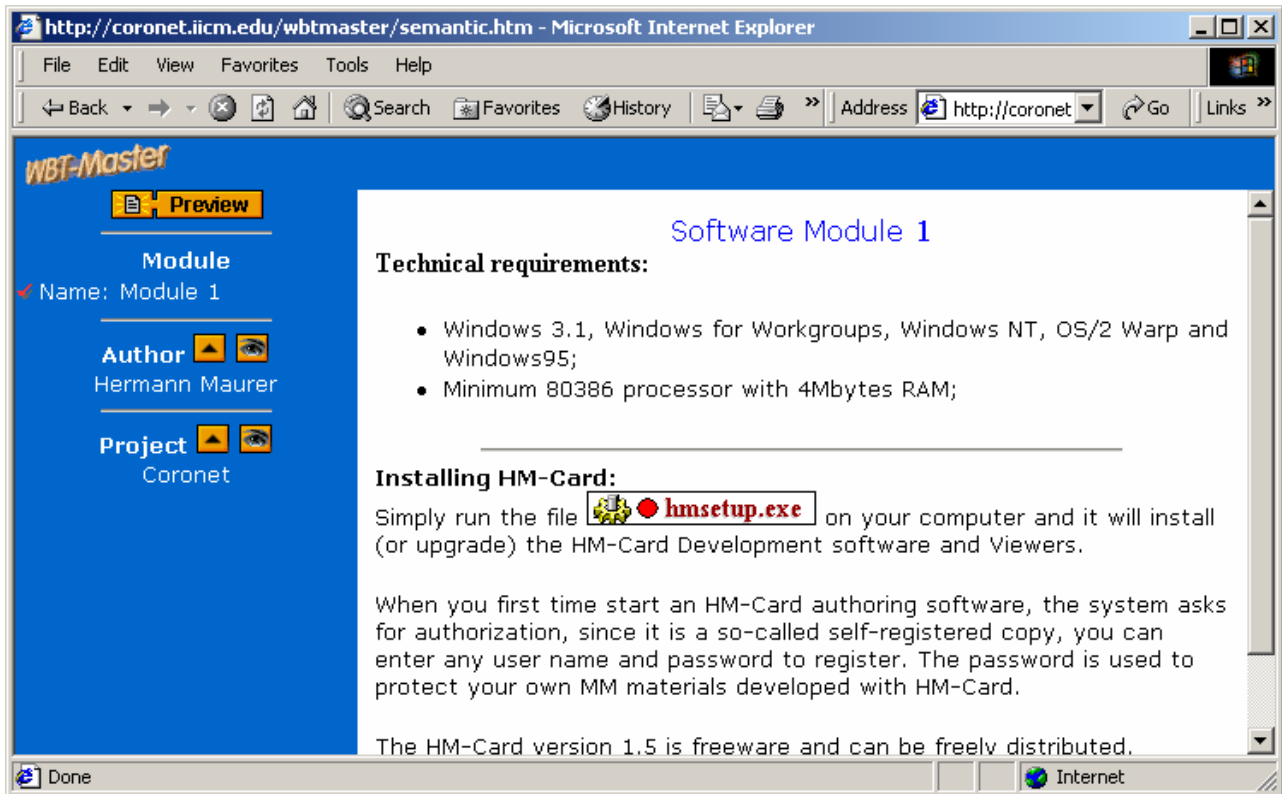


Figure 3: Browsing a Knowledge Domain

Thus, browsing a knowledge domain does not require any additional knowledge besides understanding of the predefined structure of the domain. Users simply select a particular knowledge domain and a certain category and all existing instances are displayed by the system. Selecting a particular instance results in the start of the actual browsing of the knowledge domain.

## 6. Conclusion

The concept of the Knowledge Domains facilitates many important features such as:

- Separating of the structure and the content of the resulting hypermedia database - as shown many times before [Helic et al., 1999] [Helic et al., 1999a] [Maurer et al., 1996], the crucial point for the successful maintaining of large WWW based hypermedia databases.
- Defining of new data structuring facilities through the concept of the Knowledge Domain Schema - provides a possibility to reuse learning resources in different well-defined contexts, thus, allowing learners to have new views on the learning material collected in a WBT system.
- The concept of the Knowledge Domain Schema provides means for template-based authoring - through the template-based authoring the authoring process is considerably simplified. A wizard-like authoring system is imaginable, as it is already implemented in the WBT-Master system. Such authoring system is easy to use, and a rapid production of learning material is easy to achieve.

- The result of the authoring process is a well-structured repository of learning resources - the concept of the Knowledge Domain Schema defines a database schema that is applied by authors in the authoring process to structure learning resources. Thus, the result of this authoring process is a very well-structured database created in the accordance to a number of predefined database schemes.
- Searching in the system on a meta-level - attributes attached to learning resources, as defined by the Knowledge Domain Schema, may be comprised as the data on about the data, i.e., the meta-data. Such meta-data may be used to enhance searching facilities provided by the system, especially when they are predefined as it is the case with the Knowledge Domains.
- Browsing learning resources on a meta-level - browsing of Knowledge Domains facilitates browsing of the concepts ( comprised in semantic categories ) and relationships between these concepts ( comprised in semantic relationships between different categories ) as opposed to browsing of individual documents or learning resources. Furthermore, when browsing instances of semantic categories those instances are provided with very useful additional information contained in attribute values attached to those instances.

However, we believe that the concept of Knowledge Domains, as it is implemented in the WBT-Master, requires a number of improvements. Those improvements are mainly concerned with the concept of semantic relationships:

- It should be possible to define not only relationships of 1:n character but also relationships of m:n character; often the 1:n relationships are not sufficient to describe real-life relationships between two concepts. For instance, an instance of the category "Module" ( i.e., a software module ) may be a part of not just one "Project", but possible two or more "Projects". This facilitates the description of the concept of reuse of software modules, which is one of the basic concept in the software engineering field.
- It should be possible to define not only binary relationships, but also relationships of an arbitrary arity. Usually, to describe relationships between different concepts, we need to apply n-ary relations between those concepts. For instance, if we say that the category "Module" has an attribute "Name" of the type String, we may comprehend this fact as a relationship between concepts "Module" and "Name". If we now consider other relationships of the "Module" category ( e.g., "Project Modules" ) we end up with the category "Module" that constitutes a 3-ary relationships between categories "Name", "Module" and "Project". However, this facility should be explicitly supported by the system in order to improve the expressive power of the Knowledge Domain concept.

## REFERENCES

- [Andrews et al., 1995] K. Andrews, A. Nedoumov, and N. Scherbakov: Embedding Courseware Into Internet: Problems and Solutions. Proceedings of ED- MEDIA'95, Graz, Austria, June 1995, pp.69-74.
- [Berners-Lee et al., 1994] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret: The world-wide web. Communications of the ACM, 37:76-82, Aug 1994.
- [Cailliau, 1995] R. Cailliau: About WWW. Journal of Universal Computer Science, 1:221-230, Apr 1995.
- [Helic et al., 1999] D. Helic, H. Maurer, N. Scerbakov: Introducing Hypermedia Composites to WWW, Journal of Network and Computer Applications, 22, 1, Academic Press, 19-32
- [Helic et al., 1999a] D. Helic, H. Maurer, N. Scerbakov: Authoring and Maintaining of Educational Applications on the Web, Proceedings ED-MEDIA'99, AACE, Charlottesville, USA, 1792-1797
- [Helic et al., 2000] D. Helic, H. Maurer, N. Scerbakov: Web Based Training: What Do We Expect from the System. Proceedings of ICCE 2000, Taiwan (2000), 1689-1694
- [Fernandez et al., 1997] M. Fernandez, D. Florescu, J. Kang, A. Levy and D. Suci: Strudel: A Web-sit management system. Proceedings of ASM SIGMOD'97 Tucson, Arizona, May 1997 pp. 549-552.
- [Maurer et al., 1996] H. Maurer (Ed.): HyperWave: The Next Generation Web Solution. Addison-Wesley Longman, London. 1996.