RC23880 (W0602-132) February 15, 2006
Computer Science

IBM Research Report

Semantic Composition of Learning Object Collections

Robert G. Farrell, Brian F. White
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Semantic Composition of Learning Object Collections

Robert G. Farrell[1] and Brian F. White[1]

[1] IBM T J Watson Research Center
Hawthorne, NY USA
{robfarr, bfwhite}@us.ibm.com
http://www.research.ibm.com

Abstract. Learning object collections may be developed by different people, teams, or organizations using different metadata. This paper explores the issues in composing collections by combining content and metadata. We describe an implementation of semantic composition in the context of a particular system, Dynamic Learning Experience, a web-based application that allows learners to query, assemble, and organize learning content into hypermedia learning paths.

1 Introduction

It is widely recognized that complexity can be managed through composition of simpler units according to a set of well-defined rules. For example, software system complexity can be managed through composition of component building blocks. Principles of composition have been applied to object-oriented programming [1], web services [2], electronic circuits [3], and other domains. We believe that a similar approach can be taken to the complexity of hypermedia.

We have explored composition of hypermedia in the context of learning objects. A learning object can be digital, such as a document, a video, or an on-line course, or can be non-digital, such as a conference or mentoring activity. In this work, we focus on the composition of collections of digital learning objects. A learning object (LO) is a unit of learning content with a specific learning objective, often used as a building block in designing and assembling courses. The learning objects that make up collections are tagged with metadata, such as topic or level of difficulty, and they are stored in a repository.

The composition of learning object repositories is less complex than the composition of other hypermedia for two reasons. First, learning objects have metadata [11] that provides semantics. Second, learning objects are self-contained, so composition of repositories does not involve the problem of creating links within learning objects. However, the common metadata schema only provides semantics for learning objects within a given repository, so semantic composition among repositories is required.

A Dynamic Learning Environment

We have explored learning object repository composition in the context of the Dynamic Learning Experience (DLE) system [4][5]. This section describes the user experience, system architecture, and data model.

User Experience

Using DLE, learners are able to generate their own hypermedia learning paths composed of learning objects retrieved from a repository. Users select a repository (e.g. 'Websphere Education') and enter a query ('configuration'), desired course duration , and desired depth of study (see Figure 1). Advanced search options let users restrict the assembled custom course to a particular type of material and level of difficulty.

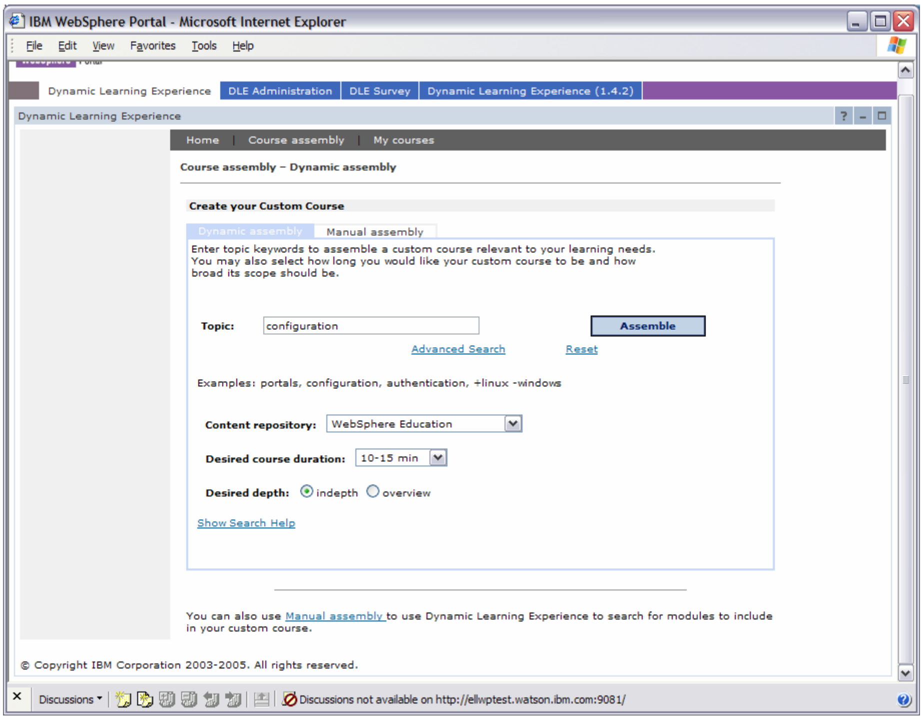


Figure 1: The Course Assembly page

The system retrieves sections of reference books, units of course material from classroom presentations, videos, and other modular learning content, and then assembles and sequences the search results into a custom course (a linear hypermedia learning path personalized to the user) consisting of learning objects appearing as numbered lessons (see Figure 2). Users can drag and drop learning object lessons to reorder them and perform other customizations. When done, they can play the custom course immediately or start the custom course from any lesson.

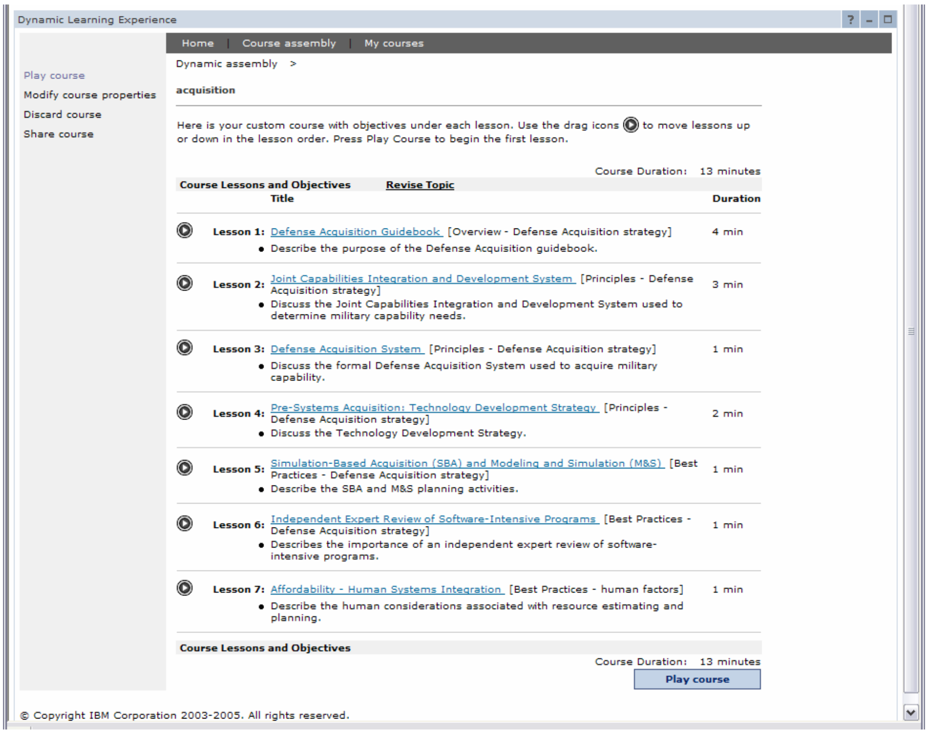


Figure 2: A Custom Course

System Architecture

The Dynamic Assembly Engine requires that learning objects are tagged with a topic classification, an intended use for the object in an instructional context (e.g., introduction), and a time estimate for reading the content or performing a described activity. The type of learning material (e.g. slides), difficulty level, and target audience can also be tagged as metadata.

DLE consists of a Search Engine, a Dynamic Assembly Engine, and a Course Player (see Figure 3).

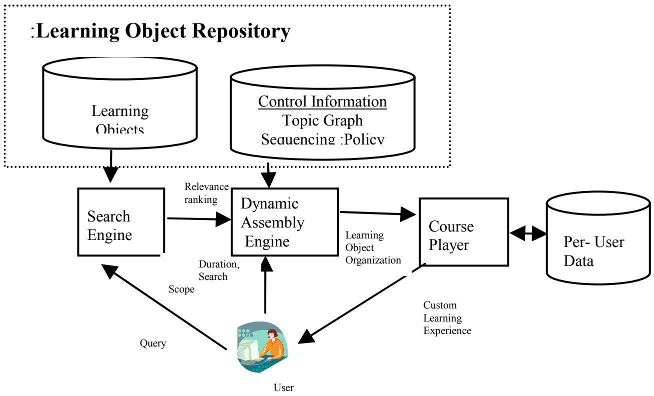


Figure 3: Dynamic Learning Experience architecture

The Search Engine returns a relevance ranking of search results from the repository of Learning Objects. The Dynamic Assembly Engine: 1) maps these search results to nodes in a Topic Graph based on which topic each learning object is tagged with, 2) computes the relevance for each topic node based upon the mapped learning objects, and then 3) uses both the relevance for each topic node and the relationships between topic nodes to find a best path through the graph [13]. The learning objects on the path are then organized into a custom learning experience according to a Sequencing Policy. For example, the policy may specify that learning objects should be organized by a default topic order or instructional use order.

The Topic Graph has nodes for topics and edges for topic relationships. Nodes are encoded as entities in the Resource Description Framework (RDF)[20]. Relations between entities are declared using OWL properties[21]. Topic graphs are similar to topic maps [9], concept maps [7], and Brase and Nejdl's ontologies [6]. Topic Graphs can be a simple list of topics, a hierarchical taxonomy of topics, or a graph of associations. The Topic Graph and the Sequencing Policy can be customized for each collection or learning objects.

Learning Object Repository Composition

A learning object repository is a collection of learning content and associated learning object metadata. To enable dynamic assembly, DLE repositories include not only learning objects with metadata, but also control information in the form of a set of declarative specifications that provide additional semantics to the metadata schema and also provides behavioral information for how to assemble sequences of learning objects for presentation.

The Fundamentals of Repository Composition

The most fundamental kind of learning content repository in DLE is a base repository. It is concerned with a specific well-understood domain, and is atomic in the sense that it does not refer to information in any other repository, though two base repositories may have learning objects in common. A composite repository is created by combining learning objects from two or more other content repositories, and inherits aspects of their semantics, which to say it shares some part of the control information of the repositories it draws from. Base repositories are usually defined manually and serve as building blocks for composite repositories that are created automatically or semi-automatically.

We have created a set of algorithms for automatic combination of content repositories. If portions of several content repositories are combined, the algorithms will aggregate the learning objects and merge their control information, creating a completely functional new composite repository. However, the resulting content collection may still need to be refined manually. We explain below the complexity involved in the merging decisions, so it becomes clear when and why human intervention is needed.

Issues in Repository Composition

We have identified the following issues when composing repositories:

1. **Scaling** – Some metadata may be tagged in a relative manner. For example, the difficulty level may be judged relative to other content. When combining repositories, these values may have to be rescaled.
2. **Inconsistent methods** – Some metadata may be computed automatically, but the method may be different for different repositories. For example, the duration may be estimated based on average adult reading time for one repository but may be based on a complexity measure for another repository.
3. **Semantic mapping** – Some metadata values may mean the same thing, but simply be expressed differently in different repositories. For example, a *developer* audience in one repository may be equivalent or nearly equivalent to a *programmer* audience in another repository.
4. **Semantic distance** – Two repositories may be semantically distant, requiring some “bridging” concepts. For example, both may have a classification that places the learning objects in a taxonomy, but these taxonomies may have entries that cannot be directly mapped, but they may be related indirectly through a more encompassing taxonomy.
5. **No cross-repository semantics** – In some cases, an encompassing semantics may not be possible, so repositories may need to be partially disjoint.

When combining control information in DLE repositories, topic graphs are spliced together. This is straightforward if nodes of each graph have unique identifiers drawn from a common namespace. However, because repositories are developed relatively independently, this may not be the case. We investigated mapping the local topics created for a learning object repository to a larger cross-domain taxonomy using automatic text classification [16]. However, we found that the cross-domain taxonomy is typically at a higher level of abstraction. We also investigated finding connections between topic words and phrases using WordNet[17], a cross-domain thesaurus.. However, many of the learning object repository topics used in our corporate learning content repositories involved highly technical terms not present in WordNet. For some repositories with sufficient numbers of text-based learning objects per topic (taken from sections of technical books), we were able to train a regularized linear classifier [18] to automatically categorize with a sufficiently high level of precision[19] Text-based learning objects can then be assigned to topics in the composite repository.

Merging Behavioral Characteristics

When composing repositories, it may be important to merge the behavioral characteristics of the two repositories. For example, if rules are specified for how to present collections of objects, the rules must be combined. If rules are not independent they may not be able to be easily combined.

DLE relies on a default order for topics that partially determines the order of presentation of learning objects. For the purposes of composition, each default order is treated as a linear order (a list). The two lists are compared and merged. Since the two lists may have topics in common that are not consistently ordered between the two, such inconsistencies are resolved by removing inconsistent topics from one list until the remaining common topics are in the same relative order in both lists. The two lists are then aligned with each other to create a merged ordering. If the sequencing policy is set to topic order, learning objects will be sorted according to their position in this new list.

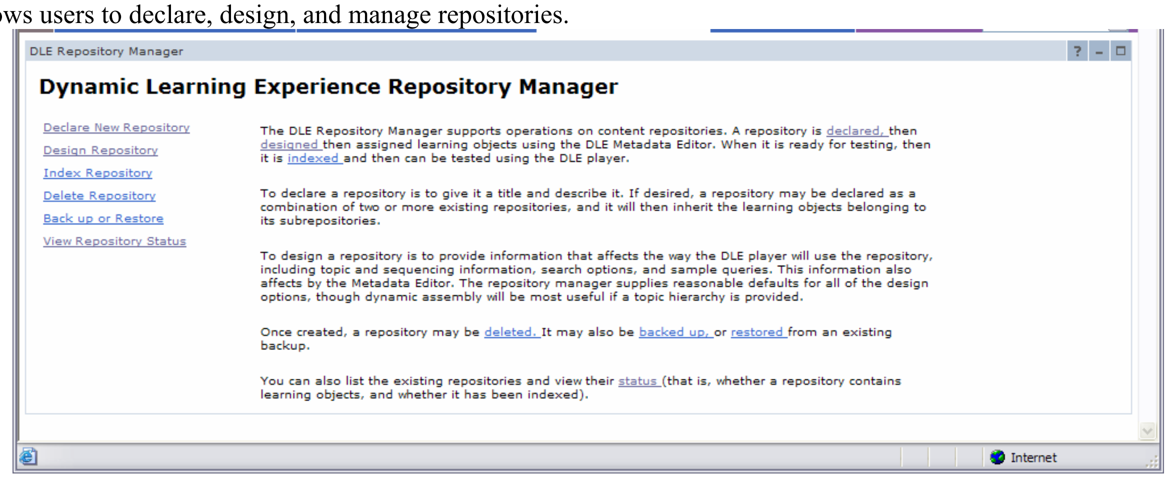
For example, if one list specifies the topic ordering A, C, E, D and another the ordering A, B, D, E, then the inconsistent ordering of D and E may be resolved by removing E from the first list, yielding A, C, D. Comparing the lists, we know that B and C both fall between A and D, and, although their relative ordering is undetermined. Thus we can (non-uniquely) generate the ordering A, B, C, D, E. Using topic order as a sequencing policy, learning objects whose metadata associates them with topic A would be presented before learning objects with topic B, and so on.

Other control information is also merged, such as the schema vocabulary. That is, in creating a query in order to assemble a custom course, the user may wish to constrain the query by using attributes such as instructional use; for example, by choosing the appropriate value for learning resource type, one could restrict the query to slides. The choice of attributes and attribute values may vary according to the semantics of a given repository and the kinds of learning objects it contains.

When two repositories are composed, the schema vocabulary of the composite is the union of the attributes of the constituent repositories. The values for an attribute whose name occurs in both repositories is the union of its values in the two repositories. If values are not taken from a standard vocabulary, they may have to be semantically mapped to determine synonymy.

The DLE Repository Manager

We have created a web-based application, the DLE Repository Manager, that allows users to declare, design, and manage repositories.



Defining a Repository

In the declare stage, users can declare repositories as a base or composite and if composite, they can select the included repositories. They can also input Dublin Core Metadata, such as a title and description.

The system allows complete control over what parts of repositories are combined to create a composite repository using set-theoretic operations. The smallest building block is a resource and consists of an individual directory or file. A resource is associated with a particular source repository. Resources are bundled together in groups, which are not themselves linked to a given repository, and may in turn be defined in terms of other groups. A content repository may be defined in terms of one or more groups, one or more resources, or one or more included repositories, or a combination. A content repository may be wholly included or included in part. We may specify that only some part of a repository should be included, or that all of a repository should be included except for some excluded set of objects. The objects that are included or excluded are specified as groups.

Defining a Metadata Schema

In the design stage, users can define the metadata schema for their repository. The schema constraints on values for elements such as Difficulty and Intended Use are managed by editing rules. For example, difficulty values must come from the tokens specified in the IEEE Learning Object Metadata standard, but users can specify that only a subset of values should be used (e.g., easy, medium, and difficult).

Learning object metadata values specified in the DLE Repository Manager are then used to configure authoring tools for content developers wanting to create learning objects for a given repository.

Discussion

Shared learning content repositories foster resource sharing and retrieval, but only provide a global semantics such as subject taxonomy or other fixed metadata. Educators must understand the required metadata when submitting resources. There is no attempt to merge independently-developed repositories that do not share a common semantics.

Learning object repositories using the Sharable Content Object Reference Model (SCORM) [12] can be federated using the Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) [14]. The federated architecture allows repositories to be registered along with metadata for individual learning objects. Implementations of CORDRA have imposed a strict metadata schema on participating repositories, however, and there is no ability to support semantic descriptions for dynamic content assembly or to merge repositories.

The Adaptive Course Construction Kit (ACCT) provides for composition of adaptive courseware using a narrative structure [22]. This system stresses reuse through definition of a reusable ontology and a library of adaptation techniques. However, this system does not support automated repository composition.

How much of the composition process can be automated? In the case of DLE, we are able to automatically generate composite repositories that may be immediately used. Assuming that the content itself does not change after composition, it also appears that there is limited need for manual correction of the control information of the composite repository. First, the control information is tied to tags appearing in the learning object metadata – one cannot change the set of topics ad hoc, for example, without changing the content itself. Similarly, the schema vocabulary cannot be added to without changing the learning object metadata, and reducing it will only limit access to the existing information. Second, the specific kinds of control information used permit well-defined composition operations, barring name clashes. For example, forming the union of the schema vocabulary sets of constituent repositories will yield sensible results. One might want to refine them, if some elements of the union are not very well-represented in the composite repository, but this is not a fundamental issue. It is reasonable to splice topics graphs together, providing the constituent graphs are semantically sound. Topic orderings for the instructional order of content can be combined in a consistent way.

While the composition of the learning objects themselves, and their metadata, does not itself present an issue in DLE, there are some subtle semantic issues that may arise. For example, the difficulty level of a learning object as assigned within a constituent repository may not be quite the same as judged for a composite repository in which it appears. One intended use tag (e.g., "scenarios") may subsume or overlap with another (e.g., "case studies"). We are exploring these issues as we refine the algorithm for repository composition and the support for manual review and editing.

Conclusion

Automatic composition of content involves integrating content, metadata, semantics, and behavior. We have described some of the issues in composing learning object repositories, including combining learning content, metadata schemas, and topic graphs. This work provides one example of how the complexity of hypermedia can be managed by using principles of composition that have been successful in object-oriented programming, web services, and other domains.

References

1. Nierstrasz, O. and Meijler, T.D. Research Directions in Software Composition. ACM Computing Surveys, 1995.
2. Srivastava, B. and Koehler, J. Web Service Composition-Current Solutions and Open Problems. ICAPS 2003 Workshop on Planning for Web Services, 2003.
3. Koegst, M., Garte, D. , Conradi, P., and Wahl, M.G. A Systematic Analysis of Reuse Strategies for Design of Electronic Circuits. DATE, 1998.
4. Farrell, R. et.al. "Learner-driven Assembly of Web-based Courseware". *Proceedings of E-Learn 2003* (Phoenix AZ, Nov 2003).
5. Farrell, R. et.al. "Personalized Just-in-time Dynamic Assembly of Learning Objects". *Proceedings of E-Learn 2004* (Washington, D.C., Nov 2004).
6. Brase, J., Nejdl, W. Ontologies and Metadata for eLearning, Handbook on Ontologies, pp. 579-598,Springer Verlag, 2003.
7. Nilsson, M & Palmér, M., (1999) Conzilla - Towards a Concept Browser, CID-53, TRITA-NA-D9911, Dept. of Numerical Analysis and Computer Science, KTH, Stockholm, Sweden. http://kmr.nada.kth.se/papers/ConceptualBrowsing/cid_53.pdf
8. Noy, N.F. and Musen, M.A. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In Proceedings and the National Conference on Artificial Intelligence (AAAI), 2000.

9. Park, J. (2003). Topic Maps, the Semantic Web and Education. In J. Park (Ed.), XML Topic Maps: Creating and Using Topic Maps for the Web (pp. 507-529), Boston: Addison-Wesley
10. Peng Wang, Baowen Xu, Jianjiang Lu, Yanhui Li, Dazhou Kang, Theory and Semi-automatic Generation of Bridge Ontology in Multi-ontologies Environment, Lecture Notes in Computer Science, Volume 3292, Jan 2004, Pages 763 – 767
11. IEEE 1484.12.1-2002 Learning Object Metadata v1 Final Draft,http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
12. Dodds, P. and Thropp, S., SCORM 2004 2nd Edition Overview. http://www.adlnet.org/scorm/history/2004/
13. Farrell, R., Liburd, S.D., and Thomas, J.C. Dynamic Assembly of Learning Objects. WWW 2004.
14. Rehak, D., Dodds, P., and Lannom, L. (2005) A Model and Infrastructure for Federated Learning Content Repositories. In Proceedings of the Interoperability of Web-Based Educational Systems Workshop (Volume 143 of CEUR Workshop Proceedings, Technical University of Aachen).
15. Thomas, R. and Farrell, R.G., "An Experimental Investigation on the Effectiveness of Individualized Web-based Learning based on the Dynamic Assembly of Learning Objects" IBM Technical Report RC23338.
16. Gates, S. C., Teiken, W. and Cheng, K.-S. F. 2005. Taxonomies by the Numbers: Building High-Performance Taxonomies. In Proceedings of the Fourteenth ACM Conference on information and Knowledge Management (Bremen, Germany, October 31 - November 01, 2005). CIKM '05. ACM Press, New York, NY.
17. Miller, George A.; Beckwith, Richard T. Fellbaum, Christiane D.; Gross, Derek; and Miller, Katherine J. (1990). "WordNet: An on-line lexical database." *International Journal of Lexicography,* **3**(4), 235-244.
18. Zhang, T., Oles, F.J., (2001). Text categorization based on regularized linear classification methods, Information Retrieval, 4, 5-31.
19. Kannampallil, T. G. and Farrell, R. "Automatic Learning Object Categorization for Instructional Usage using an Enhanced Linear Text Classifier". The Second Internaitonal Conference on Knowledge Management (ICKM 2005).
20. Resource Description Framework (RDF) Model and Syntax Specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, W3C Recommendation. February 22, 1999.
21. OWL Web Ontology Language overview, http://www.w3.org/TR/2003/CR-owl-features-20030818/.
22. Dagger, D., Wade, V., and Conlan, O. (2005). "Personalisation for All: Making Adaptive Course Composition Easy" *Journal of Educational Technology and Society,* **8**(3), 9-25