

## A Model for Distributed Curriculum on the World Wide Web <sup>+</sup>

Tom Murray

### Abstract:

Educational material abounds on the Web, but still most designers of instructional material must start from scratch. Emerging interoperability standards for educational resources will accelerate progress toward reusability of this material. We describe a framework called Model for Distributed Curriculum (MDC) that uses a topic server architecture to allow one Web-based tutorial to include a specification for another tutorial, a “virtual tutorial”, where the best fit to this specification will automatically be found at run time. This allows a designer of instructional material to re-use other Web-based tutorials without having to know what tutorials will exist at run time, i.e. without having to search and analyze material on the Web. Key to MDC is organizing Web-based curriculum into a “topic space” rather than a “page space,” and making use of emerging metadata standards for educational resources to describe the pedagogically relevant properties of Web-based tutorials.

### Keywords:

adaptive curriculum, interoperability, Web-based learning, intelligent tutoring systems

### Commentaries:

All JIME articles are published with links to a commentaries area, which includes part of the article’s original review debate. Readers are invited to make use of this resource, and to add their own commentaries. The authors, reviewers, and anyone else who has ‘subscribed’ to this article via the website will receive email copies of your postings.

Tom Murray, *Computer Science Department, University of Massachusetts, Amherst, MA 01003 & Computer & Cognitive Studies Program, Hampshire College, Amherst, MA 01002, U.S.A.*  
*TMurray@cs.umass.edu | www.cs.umass.edu/~tmurray*

<sup>+</sup> *This research is based on work supported by the National Science Foundation and the Advanced Research Projects Agency under Cooperative Agreement No. CDA-940860.*

## **1. Introduction**

The World Wide Web's potential to become a universal medium for educational resources and educational agents has been recognized since its birth, but only recently have the necessary pieces begun to fall into place. Just as standardization of document structures and media formats has been key to the Web's current success, new standards are needed to allow Web-based educational agents and resources to interoperate. There is already a large amount of instructional material on the Web, most of it in the form of multimedia-embellished HTML documents. More highly interactive content is being produced every day using Dynamic HTML and Java. Knowledge-based, generative, and intelligent instructional environments (all of which I will refer to as "intelligent tutors") are just beginning to appear on the internet (Brusilovsky, Schwartz and Weber, 1996; Stern, Woolf and Kurose, 1997; Kiyama, Ishiuchi, Ikeda, Tsujimoto, and Fukuhara, 1997).<sup>1</sup> While the sophistication of Web-based instructional materials is increasing, it is still catching up to the levels of functionality and performance that have been achieved in stand-alone instructional programs and CD-ROMs. Web-based instruction will likely continue to lag behind for three to ten years until the bandwidth of the average Web connection increases dramatically. The real strengths of the Web over stand-alone applications, i.e. its standards-based uniformity and the potential for widely distributed resources to interoperate with each other, have been only marginally tapped.<sup>2</sup> Current educational use of the Web is primarily limited to simple hyperlinks between content pages, and to home pages and other "portal pages" which organize a set of related links. True interoperability and reusability of these educational resources is practically nonexistent.

The amount of effort being put into Web based educational material is substantial and growing rapidly. Since this material is expensive to generate, it is generally agreed that enabling easy re-use and re-purposing of this material is a high priority. The problem of re-usability of formatted text and media resources has been largely solved on the Web with HTML, MIME types and other data format standards. Such reusability does not exist at the content, tutorial, or pedagogical levels however. There is no easy way for an instructor or learner to find a tutorial on the Web that addresses particular learning and pedagogical goals. Current Web searches are

<sup>1</sup> *Intelligent tutors are much more sophisticated than multimedia hyper-linked documents, in that they can include teaching strategies, device simulations, coaching, and problem solving expertise in the domain of instruction. Sophistication at this level can currently only be achieved using a client-server approach, wherein a dedicated Web server uses CGI programming to generate Web pages dynamically.*

<sup>2</sup> *The Web also provides new opportunities for communication and collaboration among geographically and temporally dispersed participants. These capabilities, important to the overall picture of technology-enhanced learning, are outside the scope of this paper.*

typically frustrating hit-or-miss endeavors because most search engines search the contents of Web pages and try to infer what they are about, what their goal is, or how they relate to other pieces on content. Fortunately, emerging Internet standards for metadata (information which describes an electronic document or resource so that it can be more efficiently located and re-used) should allow for more exact and “intelligent” Web searching. With appropriate metadata standards in place, assuming enough metadata for educational resources is created, it will be much easier and more efficient to search the Web for content that satisfies particular constraints, such as “teaches about plate tectonics at a high school level,” or “short tutorial about cloning containing pictures and examples.”

This paper describes a method for specifying virtual (i.e. hypothetical) tutorials by specifying the learning goal (or topic) and pedagogical constraints (e.g. length and instructional style) of the desired tutorial, rather than specifying an actual tutorial (nominally a URL). The virtual tutorial (or tutorial “request”) is processed to return an actual tutorial at run time. This method, called Model for Distributed Curriculum (MDC), makes use of emerging metadata standards to allow Web based tutorials to invoke other Web based tutorials.<sup>3</sup> MDC includes a scheme for representing the elements of a distributed curriculum (tutorials, objectives,, courses, etc.) and managing a learner’s involvement with these elements. By “distributed” we mean located at arbitrary locations on the Web, and authored by multiple authors. Tutorial interoperability and automated record keeping of learner performance will enable educational institutions to implement the types of individualized (non-age grouped) learner-driven instruction, and project-based learning that has long been acknowledged as beneficial, but has not yet been cost effective (Osin and Lesgold, 1996).

## **2. Learning Technology Components and Standards**

To provide a context for the description of the MDC, I will briefly describe the major components of computer based tutorials, and the proposed IEEE standards that aim to facilitate interoperability between tutorials and between components of tutorials.

The field of intelligent tutoring systems uses a four-part conceptualization of computer-based

<sup>3</sup> *This paper describes the state of the art and potential uses of technology in an area in its early infancy. The model described here has been only partially implemented, and has not been evaluated in any way. I would not normally offer work at such an early and speculative stage for publication, but the editors agreed that it was important to contribute to the field’s nascent dialog on the characteristics of the research space, and to further articulate the important research questions and issues. The bulk of my research with the East/West Consortium, on which this special issue is focussed, dealt with authoring tools and is fully described in Murray (1998).*

instruction (Wenger, 1987) that is useful for our discussion (even though our context is not limited to “intelligent” tutors). The four parts are: 1. a model of content and domain expertise (which can contain an expert system and a curriculum structure), 2. a model of teaching (and communicating) expertise, 3. a user or learner model, and 4. a user interface. In research on generic architectures, and authoring tools (Major and Reichgelt, 1992; Murray, 1996a; Van Marcke, 1992) there is an effort to make these four components as modular and independent of each other as is feasible. In theory, each of these elements can exist as autonomous resources on the Web, used by multiple tutorials and operating independently of any single tutorial, as described below:

**Autonomous knowledge base**, or any repository of instructional content, including a domain model, multimedia data base, curriculum model, or expert system.

**Autonomous learner model/record:** Could include: 1) a record of learner behavior, activities, scores, etc.; 2) a record of general information about the learner (personal information, standardized test scores, academic transcript, etc.) and/or 3) a model containing inferences of what the learner knows or doesn't know, based on learner performance and behavior.

**Autonomous instructional strategies:** General specifications of when and how to perform communicative and instructional actions. For example, “Socratic method” or “programmed learning” strategies. Though strategies can be domain independent, some will be less general, as in specific language instruction methods, or a method for teaching diagnosis of automobile parts.

**Autonomous interfaces, simulations, and widgets:** The interface may be simple, as in an HTML browser, or more intelligent, having sophisticated rules for how and where to display information. Simple widgets (such as sliders, buttons, graphs) and complex simulations (such as a simulated car engine or aquatic ecosystem) can be invoked and reused by multiple tutorials (see Steven Ritter's article in this issue).

Except in research projects or very limited contexts, such autonomous resources are not yet available for general use, but current standardization efforts (see below) promise to accelerate their emergence. The most complete use of these autonomous Web-based resources would be a tutorial that accesses a knowledge base for the content it presents, stores and accesses learner information with the user model, taps a strategy resource for pedagogical methods, and/or uses a standard interface or UI widget set. But less obvious interactions among the four components are possible, such as a user model accessing a knowledge base for information needed to analyze a learner's behavior.

The networking technologies needed for interoperability of tutorials and autonomous educational resources already exist on the Internet. What is missing are standards (or commonly used de facto standards) allowing instructional resources to exchange both content (such as instructional material and learner records) and control information (so that one resource can ask another to do something). Such standards are now being developed by several groups, under the aegis of the IEEE Learning Technologies Standards Committee (LTSC, <sup>4</sup> and see Ritter and Suthers, 1997). LTSC's goal is to "develop technical standards, recommended practices, and guides for software components, tools, technologies and design methods that facilitate the development, deployment, maintenance and interoperation of computer implementations of education and training components and systems." <sup>5</sup> The LTSC is currently composed of about a dozen working groups and study groups, including:

1. **Learner Model.** To characterize a learner and his or her knowledge/abilities, such that other systems can access and update this information in a standard way. The standard will also address issues of privacy and security.
2. **Tool/Agent Communication.** To establish a protocol to govern interaction between software tools and instructional agents.
3. **CBT Interchange Language.** A standard format for computer-based courseware that will allow courseware content to be easily ported from one courseware authoring system to another.
4. **Learning Objects and Metadata.** A metadata standard that will enable learners, instructors, and computer systems to search, evaluate, acquire, and utilize modular content and other instructionally relevant resources.

<sup>1</sup> *IEEE Learning Technology Standards Committee (LTSC) (P1484)*  
<<http://www.manta.ieee.org/p1484>>

<sup>2</sup> *All descriptions in this paper of the IEEE LTSC standards refer to draft standards that are still being modified. The LTSC standards process is young, and all of its working groups and documents are in early stages. The list of standards areas shown here is an incomplete list of areas in which members have begun to articulate the problems and opportunities, define the scope and purpose of a proposed standard, and in some cases put forth initial draft standards documents. There is no guarantee that any of these documents will become mature enough to be standards and be voted into acceptance. Even if a standard is accepted by IEEE membership vote, there is no guarantee that it will prove useful and profitable enough to various stakeholders to be widely adopted.*

5. **Learning System Architecture.** A conceptual model (“reference model”) that identifies the main components of learning software and what information these components need to share.
6. **Task Model.** A standards for describing tasks or job descriptions so that, 1) jobs/tasks can be matched to learners who have the correct knowledge and skill (according to a learner model), and 2) a person wanting to perform a task can self-train by finding the learning resources necessary for the job.

Paepcke, Chang, Garcia-Molina and Winograd (1998, p. 39) affirm that “an appropriate standard that is widely adhered to provides a powerful interoperability tool [and] encourages infrastructure investment.” However, they note that standards can also “infringe on site autonomy [and] often end up being complex combinations of features reflecting the interests of many disparate parties.” They discuss several metrics for evaluating interoperability standards: autonomy, cost, ease of contribution, ease of use, task complexity supported, and scalability. Often these goals will conflict with each other and tradeoffs need to be made. For instance, designing a standard for greater task complexity will usually result in decreased ease of use. We discuss design metrics for one of the proposed standards in the Metadata Vocabulary Section.

The remainder of this paper is concerned with tutorials as autonomous resources or agents that can utilize (“call” or “jump to”) each other.<sup>6</sup> The model proposed in this paper uses the Learning Objects and Metadata (draft) standards mentioned above to characterize tutorials and topics in a distributed curriculum. Tutorials are treated as “black boxes”---it is not relevant how they work or whether their source for content, learner information, etc. is external or internal. The model proposes external autonomous knowledge bases and student models that mediate *between* tutorials. It includes a knowledge base that knows the relationship between the topics covered by different tutorials, and a user model that stores data from sessions with multiple tutorials. We are not concerned here with how each tutorial is implemented, but in how each one can be encapsulated to allow it to interact with others. For example, a simple hyper-text document explaining the inner ear should be able to link to a sophisticated intelligent tutor that teaches about ear acoustics, and an intelligent tutor teaching aircraft maintenance procedures should be able to invoke an HTML page about pneumatic pumps. If it were just a matter of pointing to a known tutorial (via a URL) and jumping to it and displaying the contents in a Web browser, this would be trivial. But our goal is to support “intelligent” invocation of Web-based material, which includes mechanisms that will find the right tutorial on the Web even though the author does not know what that tutorial is.

<sup>6</sup> *For our purposes a “tutorial” (or “tutor”) is any resource you “go to” to learn something or to get information, which in its most degenerate form could be a simple Web page, but it could also be a highly interactive intelligent tutor or an interactive learning environment.*

MDC is nominally about how one tutorial can invoke another, but it actually allows a tutorial to be found and invoked by anything. The nominal case is when one tutorial invokes another tutorial to teach something it can not teach (as in jumping to material that is remedial, advanced, or tangential with respect to the first system), or to teach a topic it can teach but in a different depth or teaching style. But MDC allows an ostensibly *non-instructional* Web page (e.g. a product description page) to provide an intelligent link to a tutorial that provides instruction on an important concept. We also include the case of a learner having a learning goal and initiating the search for an appropriate tutorial on the Web. Thus the focus is not on where the request for additional tutoring originates from (a learner, an information source, or another tutorial) but in how to formulate that request, find the appropriate tutorial, and facilitate the connection to that tutorial.

### 3. Separating Topics from Tutors

To reiterate, our question stated most generally is:

- How can Web-based instructional systems of any type be encapsulated so that other instructional systems can interoperate intelligently with them?

As mentioned above, the trivial case of inserting a URL link into a document needs no discussion. We are interested in situations where there is an instructional goal (posted by the learner or by a program) and the resource for achieving that goal is not explicitly stated or known. In a sense, we would like information (called meta-content or metadata, see Guha, 1996), attached to tutorials which describes what and how they teach so that a search engine can locate the appropriate tutorial on the Web. To facilitate the reasoning required in such a search we propose that the search be done in a space of instructional goals or objectives. We propose objects called “topics” for abstract representations of knowledge and learning goals. Topics can refer to any type of knowledge or learned behavior, including concepts, facts, and procedures, and there are no restrictions to the size or level of granularity of the knowledge they refer to. Example topics include “tectonic plates,” “automobile gas pump repair,” “naming the state capitals,” “ability to identify the five main types of poisonous mushrooms,” “the legislative branch of the US government,” and “working with fractions and decimals.” Topics can have sub-topics (and sub-sub topics, etc. recursively), so they can be of any scope, from a simple fact to an entire university course (though the scheme will probably not work equally well for all levels of granularity). Also, the definition of a topic does not require a clear distinction between knowledge goals (indicating something that can be *known*) and behavioral goals (indicating something that can be *done*, and usually measurably so)—a topic can refer to either or both.

<sup>7</sup> *Some instructional design theories emphasize the importance of using behavioral objectives in all structured learning. A group using MDC can certainly constrain their authors to include behavioral objectives in all topics.*

This is because the distinction between knowing and doing can be subtle in some cases, and we do not wish to require authors to work out these subtleties. Also, in some situations it will be more appropriate to author in terms of knowledge objectives vs. behavioral objectives (or vice versa).<sup>7</sup>

For now, think of topics as objects in a data base that have a name corresponding to the knowledge they refer to, and a pointer to the tutorial that teaches that knowledge. For example, an algebra tutorial may infer that a learner needs remediation in the concept of “exponents.” A knowledge base of topics is then searched for a topic dealing with exponents. The tutorial that this topic points to is invoked. Thus, the method for “encapsulating” tutorials is to create topic objects that point to them.

The simple “exponents” example above ignores a number of issues, which we elaborate below:

- **Calling parts of tutors.** A tutorial can potentially teach many topics, and use a variety of instructional methods to do so. Thus the pointer from a topic to a tutorial may contain additional information about what part of the tutorial to run, or about what presentational or instructional style to use. In one sense, topics might be thought of as indicating multiple “entry points” into tutorials.

Our scheme does not require anything of the tutorial, so any Web based tutorial resource could be used as-is. But if only a portion of the tutor’s content or capabilities is to be used, then the tutor must be created in such a way that invocations of it can be parameterized (for example by specifying subtopics, difficulty levels, overall instructional context, and/or feedback style). In the simple case of an HTML document, the topic can reference an “anchor” that points to a section of the document.

In order to make a pre-existing tutor interoperable in this scheme, one would create a number of topics, one for each bit of knowledge the tutor can teach independently, and have each topic point to the tutor with appropriate parameters that constrain what is taught. For example, in Figure 1 topics Gravity, Friction, Tension, and Force all point to Statics Tutor.

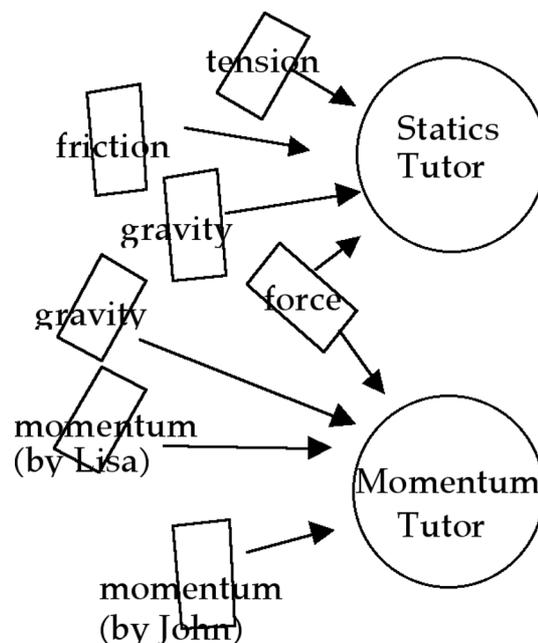


Figure 1: Encapsulating tutors with topic objects

- Topic instances vs. topic categories.** The above discussion shows how several topics can point to a tutor if the tutor teaches several things. In addition, different tutorials may teach the same topic. For example, there may be several hundred Web based tutorials teaching about “exponents.” Thus there may be several hundred topics called “exponents,” each pointing to a different tutorial. We will refer to the individual topic objects which point to a particular tutorial as topic instances and to the abstract categories, e.g. “exponents,” as topic categories. A knowledge base of topics would have only one topic category called “exponents,” and could have many instances of this topic, each pointing to different tutorials. For example, in Figure 1 there are two topic instances called “gravity” which point to different tutorials. “Gravity” in this scheme would be a topic category, and would have two instances in the knowledge base.
- Many to many mapping of topics to tutors.** Similarly, the model allows a topic instance to point to several tutors that it knows can teach it, or point to several tutors and specify the circumstances each is good for. For example, a topic instance for exponents may point to three tutors, where Tutor A is brief, Tutor B has more pictures, and Tutor C is more interactive. Therefore topic instances and tutorials have a many to

many mapping. Figure 1 shows a topic called “force” pointing to two tutorials, indicating that there is more than one way to teach about force.

- **Multiple authors.** We are assuming that the tutorials are pre-existing on the Web, and that they should not need alteration to become interoperable (an assumption that may break down for more complex, parameterizable tutors). The authors of the topic instances in the knowledge base, which I will call “metadata authors,” need not be the authors of the tutorials. Metadata authors will create topic instances that assign metadata to tutorials. Having multiple authors increases the complexity of the search space since more than one person could author a topic that describes a given tutorial. The two Gravity topics in Figure 1 could have different authors. Figure 1 shows how John and Lisa have both authored topic instances for the topic “momentum” where both point to the Momentum tutorial. However, the two topic instances may differ in some respects, such as the teaching style that the authors subjectively assign to the Momentum Tutor.
- **Term ambiguity.** Perhaps the thorniest issue involves whether one author’s definition of a topic (e.g. ‘momentum’) will be the same as another’s. We discuss methods for minimizing these and other “pragmatic” problems in a later section.

There are a number of other design and pragmatic issues involved in implementing and using MDC, which we address later. In the next section we give an example of the data structure for a topic, and in succeeding sections we elaborate on the model.

#### **4. An Example Topic and Request**

Various mechanisms for managing the interactions between tutorials are possible. The original design of MDC involved authors using a metadata authoring tool to create topic instances in an SGML compatible markup format and save these on their local sites in ASCII files identified with URLs. One or more search engines would continuously scan the Web space for topic instances and organize them. We decided against using this “Alta Vista-like” method in favor of a “Yahoo-like” method in which metadata authors use an authoring tool that registers the information with a central topic server (of which there could be more than one). The topic server processes requests for tutorials satisfying particular constraints, and returns a pointer to an appropriate tutorial, possibly with additional information. This more centralized and controlled method is easier to implement and promises to better handle some of the problems with terminological uncertainty that we discuss later.

An example of the data from a hypothetical topic instance called “Newton’s Third Law” (using an SGML-like markup syntax) is shown in Figure 2.

```

<Topic Definition>
  <name>
    newtons_third_law_easy
  </name>
  <description>
    Basic coverage of N's 3rd Law
  </description>
  <objectives>
    can identify forces in action-reaction problems
    for any two-body static situation
  </objectives>
  <relationships>
    parents = (newtons_laws)
    prerequisites = (gravity, center_of_mass)
  </relationships>
  <attributes>
    difficulty = low
    cognitive_type = qualitative_understanding
  </attributes>
  <tutor>
    <URL> http://www.um.edu/ckc/H3Rd_intro.html
    <qualifier> doctype = HTML
    <qualifier> pedagogy = pedantic
  </tutor>
  <tutor>
    <URL> http://www.cs.um.edu/ckc/Statics_tutor.cgi
    <parameter> topic = newtons_3rd
    <parameter> style = user_initiates
    <qualifier> pedagogy = socratic
    <qualifier> doctype = intelligent_tutor
    <qualifier> interactionQuality = high
  </tutor>
</Topic Definition>

```

Figure 2: Example of the data from a hypothetical topic instance called “Newton’s Third Law” (using an SGML-like markup syntax).

The topic instance has a name, a description, objectives, a list of relationships with other topic categories, a list of attributes, and a list of tutor specifications. The topic name will nominally correspond to the name of a topic instance (and topic category) in the knowledge base. The Objectives field further describes the topic by specifying what the student should be able to do to demonstrate that they ‘know’ the topic. The ‘parents’ relationship is the primary mechanism for locating the topic in the topic server’s hierarchical registry of topic categories. **Attributes** allow for terms that describe the properties of this topic instance and distinguish it from other

topic instances.<sup>8</sup> **Tutor** specifications lists tutorials that can be invoked to teach the topic. The above example has two <tutor> blocks, indicating that either an HTML document or an intelligent tutor can be invoked to teach the topic. Each tutorial may teach the topic in a different way, so “**qualifiers**” are used in the tutor specification to describe and differentiate the <tutor> blocks. In our example the qualifiers are used to specify that the first tutor has a pedantic style and the second a Socratic style. A tutorial specification may need to pass input parameters to the tutor to tell it what and how to teach. These are specified with “**parameters**”. Unlike attributes and qualifiers, which should conform to a shared vocabulary, the language of parameters is local to the specific tutor, i.e. the server just passes them along with the tutorial call, and does not need to know about the semantics.

Note that topics can refer not only to knowledge, concepts, skills, etc., but to refinements or styles of this knowledge. For example, gravity may be a topic, but so may gravity\_easy, gravity\_qualitative, gravity\_pre\_newtonian, solving\_gravity\_in\_trajectory\_problems, etc. A topic instance can refer to any knowable or learnable thing (Figure 3).

```
http://topic.server/teach_topic?
name=newtons_third_law &
difficulty = low &
cognitive_type = qualitative_understanding &
pedagogy = socratic
```

Figure 3: Example of a tutorial request generated by a tutorial to invoke another tutorial (in an HTTP query string syntax).

This specifies the instructional goal (the “name” is a topic category), and is followed with a number of **request parameters** that indicate how the topic should be taught and constrain the search for an appropriate tutorial. It is up to the topic server to: 1) search its topic knowledge base to find a topic instance that best matches the request, (2) if the topic instance has more than one <tutor> block then find the tutor that best matches the request, and (3) arrange for the indicated tutorial to be invoked. Possible arrangements include having the topic server invoke the new tutorial itself; or returning the URL and calling parameters of the found tutorial to the requester, so that the requester can decide what to do with it. Simple mechanisms may prove useful, but ultimately the server will have to be intelligent enough to deal with inadequacies, inconsistencies, and redundancies in the topic knowledge base, as discussed below.

## 5. The Promise vs. the Pragmatics

The Web is rife with educational material (though the quality of some of it may be questionable). Designers of Web-based educational applications and Web sites should be able to

<sup>8</sup> There can be any number of relationships, attributes, <tutor> blocks, and qualifiers in the specification.

re-use existing content and not have to continuously “re-invent the wheel.” Currently such instructional designers can re-use existing content by inserting a hyper-link in their document to another document. This method has two significant problems. First, the author must search the Web for a tutorial matching her needs. She must evaluate what she finds to determine that it is a good fit. This process takes significant time and is not guaranteed to produce the best fit since the entire Web can not be reasonably searched and evaluated for every topic. Second, once the tutorial has been referenced, there is no guarantee that it will exist, or exist in its original form, when a learner later uses it. The MDC topic server architecture described here mitigates these problems. First, it automatically does the searching. Second, it includes mechanisms that check for the existence and modification of Web documents. Only valid tutorials will be found and returned. Certainly such a server-based mechanism for registering and organizing information about tutorials and topics, and processing requests to return Web-based tutorials, is technically feasible. But the pragmatic problems are considerable:

1. How can instructional designers (not to mention learners) agree on a **vocabulary for topics** (i.e. the knowledge, concepts, content, subject matter areas, etc.)?
2. How can they agree upon a **vocabulary for terms** that describe the pedagogically relevant aspects of the topics and tutorials (e.g. “difficulty” and “granularity”)?
3. Even if they do agree on a set of terms for content and content descriptors, what is to say that one person’s **subjective meaning** for a term will be the same as another’s? Will my definition of “very difficult” be the same as yours? If I say I want to learn about “gravity,” how will that correspond to what you think it means to learn about gravity?
4. Even if terminology and its subjective meaning can be agreed upon, many of the terms have different meaning in different contexts. For example a tutorial can teach at a “difficult” level, but difficult for grade school students and high school students is not the same thing. How can context be accounted for?
5. And finally, if the topic server is collecting information from authors all over the world wide Web, how can we guarantee, based on our experience with current day Web search engines, that the search will not return a bunch of “garbage.” **How can we limit the false hits and misses?** How can we guarantee that, even if the data in the knowledge base indicates a good fit has been found, that it is correct and valid; and that the tutorial it points to is of good quality?

The situation could actually be worse than standard Web searching where the user knows what they are looking for and can participate in an interactive, iterative process of narrowing down a

search. In the case of distributed curriculum, the request embedded in a tutorial was authored by an instructional designer who has instructional goals that the learner is not aware of (though the learner can have a significant effect on the outcome, as explained later).

These issues are substantial, individually as well as en masse. They indicate that a fully automated topic server working on a world wide Web scale is infeasible. But still much can be done. Certainly the motivating observations are valid: there is tons of potentially reusable material on the Web; hand searching, evaluating, and linking this material is not cost effective; and a server based mechanism for distributed curriculum is technically feasible. And certainly, the need/market for good quality, easily available (i.e. Web-based) educational material is high in all sectors of society. The question then is not *whether* MDC is pragmatically feasible, but how to constrain its implementation and application to limit the pragmatic problems mentioned above. We propose several solutions, summarized here and explained below:

- Implement topic servers in **controlled or local contexts**, rather than in uncontrolled global contexts.
- Make use of emerging **metadata standards** for educational resources for standardized, yet extensible vocabularies of content and descriptors.
- Don't overestimate or over rely upon the "intelligence" of the topic server. Include human-machine negotiation steps in both the authoring process (between machine and author) and the use process (between machine and learner), in which the server presents its conclusions and its reasoning, and allows the human to affect the outcome. Thus the search and retrieval process should usually be **semi-automated** rather than fully automated.

I will discuss solution number 1 below and solutions 2 and 3 in later sections. Rather than a world-wide context such as the one in which Yahoo operates, I envision topic servers being maintained and used in fairly circumscribed contexts. Examples include: a university community, a large company's training department, a consortium of universities and businesses, a particular field such as geology, and a national home schooling project for the elementary school level. Limiting use to a particular community has several advantages. There is a much greater likelihood of establishing a common vocabulary, and having shared meanings for the terms in the vocabulary—among metadata authors, between tutorial/content authors and metadata authors, and between users/students and authors. The quality and accuracy of both the tutorials and the topic server data can be managed (as can access to the metadata). The need for and market for easy access to quality educational material exists, so if topic server technology becomes available then companies or organizations that sponsor specialized topic

servers for particular communities or educational requirements will surely materialize. Assuming there is a base-level standard for the syntax and semantics of the metadata, each community or service provider would extend or specialize the basic descriptive vocabulary. To participate in the community or use the service a user must assent to or learn its conceptual scheme.

## **6. A Metadata Vocabulary for Topics and Tutorials**

There a number of issues related to the conceptual vocabulary used in MDC. I will assume that the metadata consists of properties called “fields” and that there are constraints upon the possible values of the fields. There are three levels of conformity to discuss: a common vocabulary for the fields; a common vocabulary for the field values; and a common interpretation or meaning for the fields and the values. The fields and the values can exist within a syntactic structure, as in the “parents” property in the example above which is set within a “relationships” syntactic structure. However, for now we will ignore syntax and focus on issues of descriptive vocabulary and semantics.

In our example above we use the field terms Name, Parent, Objectives, Prerequisite, Difficulty, CognitiveType, DocType, and Pedagogy. For interoperability a scheme needs not only standard fields but also standardized data types for field values. For example, the value of Difficulty may be constrained to integers from one to five, or to a set of tokens such as “high,” “medium,” and “low.” If authors and users adopt a standard for fields and values, then what remains is the third level of conformity—that they use these terms to mean the same things. As mentioned above, constraining the author and user population to a particular community of practice is probably needed to assure reasonable uniformity. Providing adequate training and on-line help about the representational scheme will also improve matters. In addition, since, we have little precedent for communities using common vocabularies in this way, it will be necessary to employ a system that can be flexibly modified and extended to account for unforeseen user difficulties.

Creating such a conceptual vocabulary from scratch and expecting it to be adopted by any significant user population would seem to be a high risk proposition. Fortunately, an international standards group is already well underway in defining a base standard. MDC’s terminological framework will draw from the metadata standard for instructional objects and resources that is being considered by the LTSC’s Learning Objects and Metadata working group.<sup>9</sup> Metadata is information about data. In the context of the LTSC standards, the metadata is

<sup>9</sup> *This draft is based on a combination of a metadata specification developed by the Instructional Management Systems Project <<http://www.imsproject.org/>>, part of Educom’s National Learning Infrastructure Initiative, and a metadata specification developed by ARIADNE, an R&D project sponsored by the European Union with the goal of developing tools and methods for producing and supporting reuse of educational and training resources (see <<http://ariadne.unil.ch/>>).*

descriptive information about any “learning object.” Learning object is a “kitchen-sink” term meant to include instructional/learning software or documents, and components that might be used in instructional/learning software or documents, such as multimedia, courseware content, simulations, software tools, and curriculum specifications. The metadata specification includes a set of fields and a specification for the allowed values for those fields. The set of fields is intended to account for security, commercialization, life cycle, and international issues, as well as technical and educational issues.<sup>10</sup> Metadata fields from the draft standard that are relevant to our work with MDC include Description, Granularity, Pedagogy, and Learning Level (described later).

Each field is constrained as to the type of data that can fill it, and the data filling some fields has a particular sub-structure. For instance, one of the metadata fields is “Description,” which allows a free-form textual description of the learning object. But since the standard is intended for international use, the data type for the Description field is a LanguageString, rather than simply ASCII text. A LanguageString has three elements: the text itself, and a country and language specification (e.g. American/English), where the terms allowable for the language and country come from another existing standard. Several of the fields use a Taxonomy data type, which has a two part sub-structure. The first part is the concept or keyword itself, and the second is a unique identifier for the taxonomy (or vocabulary, which is a “flat” taxonomy) that the term came from. In this way the system has the flexibility to account for different conceptualizations or ontologies for how educational objects should be described. For example LCSH:greenhouse-effect might be used to indicate a term “greenhouse effect” taken from the Library of Congress Subject Headings.

In this rather applied area research and practical market-driven advancements have a particularly tight relationship. The fact that international standards are being developed, and that IMS’s draft standard may have enough stakeholders to become a de-facto standard, allows us to avoid certain arbitrary design decisions and base some of our work on a common standard. This not only means results are more likely to be relevant, but also improves the likelihood of finding significant test beds. It is still unknown whether the proposed LTSC metadata system will be useful and usable, or, more to the point, which parts or terms in the system will prove useful to what types of users. Research and practice is needed to determine what conceptual vocabularies are most appropriate, from both relevance and a cognitive/usability perspectives, to authors, learners, and other potential user groups.

Many of the fields and values used in the proposed LTSC standard are borrowed from and/or

<sup>10</sup> *Metadata fields dealing with security, commercialization, life cycle, and international issues include: language, creation/modification information, publisher, format (e.g. a MIME type), validation/authorization, operating system requirements, support service, pricing, and legal conditions.*

compatible with other standards efforts (such as the Dublin Core).<sup>11</sup> Though some of the metadata fields are standard for all electronic documents, many are specifically for educational documents and resources. Even though there are no generally agreed upon frameworks for describing the pedagogically relevant properties of educational material, the standards committee is moving ahead to propose some. Perhaps there is a belief that the creators of educational material are more likely to find common ground if they have an international standard as a focal point, even if it is far from perfect.<sup>12</sup> Thus the LTSC committee has taken a least commitment approach to specifying most of the field values. They expect modifications will follow from experience, and provide a system for extending or overriding the basic vocabulary. For example, the current draft proposes a field called “Structure” with allowed values linear, hyper-dimensional, branched, and atomic. There is no general agreement on what the organizational Structure of a resource means, and certainly no agreement that these terms serve well to cover the possible values.

The above discussion is in part to warn the reader not to balk too incredulously at the list of proposed fields and their values listed below—for several reasons. First, because the draft standard is in an early form and open to comment. Second, I believe that it is better to take a well informed guess and let people use it than wait for a system that seems to fit everyone’s needs (assuming opinions are first gathered from all of the important stakeholder groups).<sup>13</sup> Below is a partial list of the pedagogically relevant metadata fields from the LTSC draft standard.

1. Description—human-readable description of the resource. Data type: languageString (see above).
2. Concepts—list of concepts covered; from some specified taxonomy of terms in the discipline.
3. Pedagogy—Pedagogical style or approach. Possible values: inductive, deductive.

<sup>11</sup> *The Dublin Core*, <[http://purl.oclc.org/metadata/dublin\\_core/](http://purl.oclc.org/metadata/dublin_core/)> is a standard 15-element metadata set for describing any type of electronic resource, and has been adopted by library and museum information science workers. The LTSC metadata specification is a more specialized metadata set that is intended to work for educational resources and documents.

<sup>12</sup> *Decades of basic research in education and psychology have arguably failed to produce a common vocabulary for pedagogical tagging, and the chicken-egg problem will probably not go away by waiting for another round of research.*

<sup>13</sup> *The thought that a technology-oriented international standards group has declared a vocabulary enumerating, say, pedagogical styles, might seem disturbing or useless to some readers. However, the existence of a standard metadata field called “pedagogical style” that can be used to make your Web-based resources interoperable with others, and for which you can specify your own vocabulary of pedagogical styles, is more clearly universally useful.*

4. Granularity—Relative size of the resource. Possible values: curriculum course, unit, topic, lesson, fragment.
5. Structure—underlying organizational structure. Possible values: linear, hyperdimensional, branched, atomic.
6. InteractionQuality—Level of interactivity between end user and the resource. Possible values: low, medium, high.
7. Prerequisites—course, resource and/or capabilities required from the end user.
8. EducationalObjectives—Intended learning result. (Data type: not yet specified.)
9. LearningLevel—age or ability of the intended audience. Data type: a system for specifying age and/or grade level.
10. Difficulty—Possible values: low, medium, high.
11. Duration—Average time it takes to work through the resource.
12. Relation—A relationship to another resource. Data type: specifies the relationship type {ex. partOf, versionOf, basedOn, requires, isA} and the ID of the other resource.

The fields EducationalObjectives and Concepts, which refer to domain specific knowledge, skills, and behavior, seem difficult to standardize. However, the standards committee plans to make use of existing taxonomies for subject matter and content, as exist in the Thesaurus of ERIC Descriptors, the Library of Congress Subject Headings, and national standards for teaching math, science etc. The field's value will contain a unique ID for the taxonomy and the term from that taxonomy. In addition, groups that utilize the metadata system can define their own descriptive vocabularies.

The fields listed above, some of which were used in the example in the Example Topic and Request Section, are incorporated into MDC as follows (referring to the example in Figure 2): The <relationships> block of the topic definition, showing a list of relationship-type/related-topic pairs, is equivalent to the Relation field type above, but using an alternate syntax.<sup>14</sup> The Attributes block for the topic definition lists any of the metadata fields listed above, except for Relationship. Similarly, the <qualifier> block embedded in the <tutorial> block is a list of any of the metadata fields listed above, except for Relationship. Remember that the <attribute> block describes the topic, and helps distinguish it from other topics, while the <qualifier> block describes the tutorial, and helps distinguish it from other tutorials listed under the same topic definition. (See Mizoguchi, Sinita and Ikeda, 1996, and Murray, 1996b for more elaborate ontologies tailored to instructional systems.)

<sup>14</sup> Note that the LTSC Prerequisite field allows one to use various types, including free-form text description of prerequisite material, which need not be Web-based. Presumably the LTSC Relation field would be used to specify a "prerequisite" relationship between two digital resources.

Given the high uncertainty regarding the usefulness or appropriateness of many of the standards terms, the extensibility of the system is particularly important. Extensibility comes mainly from two features. First, the entire metadata set, i.e. the fields and the allowed values, can be extended or modified with a new metadata set.<sup>15</sup> Second, some of the field values specify a term and the taxonomy that the term came from. This allows a user group to define its own set of terms for the values of a field (e.g. the concepts listed in the concepts field). Researchers developing knowledge based systems and shared “ontologies” are dealing with these same issues. An ontology is a formal representation of a set of abstract objects, concepts, and relationships which are assumed to exist in a domain (Gruber 1994; Musen, 1992). An ontology can be thought of as a more sophisticated taxonomy or lexicon, which contains not only terms, but knowledge about how those terms are related. Many ontologies are designed to support agent-based knowledge sharing. They support machine inference based on the stored definitions and relationships. The above set of metadata fields and values are an ontology for describing pedagogically relevant aspects of educational resources.

Various design criteria for ontologies have been proposed (Gruber, 1994; Valente and Breuker, 1996; Swartout, Patil, Knight and Russ, 1996), including:

1. clarity and relevance (objective definitions, independent of context; terms should make sense to the intended users and not seem arbitrary or ambiguous);
2. coherence (inferences are consistent and complete);
3. extensibility and modularity (designed to anticipate revision and new uses);
4. parsimony (make a minimal “ontological commitment”; make as few claims about the world as possible)
5. theory-based (should be organized based on some guiding principles—this reduces arbitrariness and increases coherence)

These criteria are often at odds with each other, and the designer of an ontology may choose to have it excel in one criterion to the detriment of another.

How does the LTSC metadata set rate according to these criteria? An in-depth critique is not appropriate since only an early draft has been proposed. However, the fact that the framework is not based on an accepted theory of education nor are the terms generally used or agreed upon by educators does not bode well for the clarity and relevance, and theory-based criteria above.

<sup>15</sup> *A metadata extension method is currently being developed at IMS*  
<<http://www.imsproject.org>>

But a general theory in education or instructional design does not exist. The designers are likely to compensate for this by creating a fairly parsimonious and very extensible metadata set, as exemplified in the least-commitment approach to field value specification. Special purpose ontologies for particular situations can be incorporated into the core set, such as a set of metadata fields that conform to Merrill's Component Display Theory in instructional design (Merrill, 1983).

## **7. The Topic Server as a Learning Manager**

Because the topic server processes tutorial requests it can manage access to tutorials during a learning session. As mentioned, the topic server can process both tutorial-generated requests and learner-initiated requests. Thus the topic server can easily track a learner's use of tutorials, including the time spent on each one. A straightforward learner tracking feature allows for permanent records of multiple sessions and multiple learners.<sup>16</sup> With a simple extension to the scheme, tutorials can send performance data to the server so that the server can store the score, grade, or other information that summarizes the learner's performance when they exit the tutorial or finish a quiz or activity within the tutorial. Similarly, producing progress and performance reports tailored for learner, instructor, and administrator is straightforward. Though the information passing, storage, and reporting capabilities are easy to create, some thought would need to be put into defining methods for combining scores across tutorials, and for assigning scores to topics based on the performance in tutorials. Tutorials that report scores to the server would have to use compatible scoring schemes if the scores are to be combined.

The learner activity and performance record can be used to manage learning sessions. Such a record is needed to avoid sending the learner to a tutorial they have already seen. Another possible feature is to suggest to the learner that unseen prerequisite topics be visited first. More intelligence could be added to the scheme by going beyond the data of "where the student has been" toward making inferences about what the student knows. Such inferences are used to prevent both boredom (from overly simple material) and confusion from (overly advanced material). However, drawing these inferences is beyond the scope of the current model. Indeed, arbitrary levels of sophistication, drawn from the field of intelligent tutoring systems, could be added to both session management and learner modeling. Examples include adding teaching strategies such as "teach easy material before difficult," and "teach concepts before skills and principles that use the concepts." Addition reasoning can be done using the semantic network formed by the topic relationships specified in topic instances. The "parent topic" relationship creates an IsA hierarchy among topics that could be used to reason about the sequencing of topics. For instance, if a topic is to be taught, perhaps its children topics should be taught also.

<sup>16</sup> *A discussion of the important and complex issues involving privacy, security, and authenticity of user information is beyond the scope of this paper.*

As we have seen, since the topic server acts as a mediator between the learner and tutorials, it can include (or use external) learner modeling and teaching strategy modules. However, our basic model for distributed curriculum includes only minimal learner tracking and session management capabilities. These include:

- tracking (for multiple students and sessions) what tutorials (or parts of tutorials) have been visited,
- avoiding or flagging repeated visits to tutorials,
- simple session management algorithms that make decisions about the next tutorial to present (or suggest) based on topic prerequisites and learner tracking data.

## **8. Conflict Resolution and Learner Choice**

The topic server's knowledge base, containing metadata created by multiple authors, is bound to be incomplete and inconsistent with respect to the space of topics and the spaced of tutorials relevant to any particular domain. The authoring tools will help an author ensure that the name and description of a topic is consistent with other topics in the knowledge base, and will also support consistency in the topic class hierarchy (i.e. the parent-child relationships). But the attributes of a topic, the tutorials associated with a topic, and the qualifiers associated with tutorials are not restricted to be consistent with other information in the knowledge base (under the current model). In fact it is considered a strength of the scheme that the subjective analysis of different authors is represented. Thus a simple search for a tutorial satisfying particular constraints might return multiple conflicting "hits," or no complete matches at all. In such an environment topic searches need to apply some metrics to evaluate the degree of fit of each match or partial match. Our current scheme is a simple one. A point is assigned to each attribute and qualifier matched, and partial matches are prioritized according to the total number of points. We consider this method to be unacceptably primitive, but have put off designing a better one until we have experience using the system.<sup>17</sup>

Regardless of the searching, filtering, and conflict resolution algorithms used, we can safely assume that the process will not be as "intelligent" as we might wish, and that non-optimal tutorials will be retrieved. Even assuming a perfect retrieval mechanism, the learner using a tutorial will have her own needs and goals, which may not correspond to, or be able to be represented by, the tutorial request embedded inside a tutorial. Thus in MDC there is an option of returning the "best N suggestions" (where N can be specified), and listing these for the learner to choose from. The list of best matches shown to the learner will: 1) explain the goal of the

<sup>17</sup> *We plan to consult in-house researchers in the field of information retrieval regarding more elaborate matching algorithms.*

request (e.g. “the following tutors were found which most closely satisfy the criterion of: teaching about Exponents, being brief (less than 5 minutes), and using graphs”); 2) describe the quality of the match for each item found (e.g. “This item used graphs but may take 20 minutes rather than 5”); 3) allow the learner to preview the tutorial before launching into it, and 4) allow the learner to select a tutorial from the list. Not only does learner choice increase the likelihood of producing the best match, but it gives the learner more control of their learning experience, with the important affective benefits concomitant to self determination. In addition, as mentioned above, learners could make requests of the topic server directly (e.g. “I want to learn about exponents”).

## **9. A Scenario Showing MDC Use**

The quintessential MDC usage scenario from authoring to end user might look like the following.

**The Context.** A consortium of education and training organizations manages a Distributed Curriculum Service for some content area, let’s say electronics. Their goal is to develop a fully on-line curriculum suited to both bachelors’ level electrical engineers and electronics technicians. Human teachers, tutors, and guidance counselors will of course remain an important part of the academic program, as will hands-on lab work and collaborative group projects, but we will focus only on the on-line material here. The consortium is large, and thousands of Web-based educational resources already exist, including tutorials, tools, educational simulations, and informational pages related to learning about electronics and related fields.

**The software.** The group acquires shareware Distributed Curriculum software that has been successfully used by similar consortia. This software package includes: the topic server software that responds to queries initiated on the Web; a topic authoring tool for creating and managing the topic knowledge base (including a topic knowledge base “browser”); a topic request authoring tool which helps authors insert topic server requests into HTML documents (or Web based applications); an administration module that registers, tracks, and generates reports about learners who use the distributed curriculum system.

When the software package is installed one must specify the “topic ontology” or vocabulary set that will be used for topic fields. The software administrator can select from several pre-defined sets and modify these to suit particular needs. The administrator selects one called IEEE-LTSC-1, which has the fields described in a previous section, and he adds a field called Credits that records the number of academic credits given when a tutorial is used. He also changes the allowed values for the Granularity field to be “Course, Module, Lesson, Sub-Lesson.”

**Mapping the topic space and associating resources to topics.** A curriculum committee is established that creates a rough mapping of the main courses, concepts, and skills needed for the various levels of qualification (technician and undergraduate degree). Several levels of granularity are mapped, from full courses (e.g. Digital Circuit Analysis) to particular concepts (e.g. electric field generated by AC current in a wire). Hierarchical and prerequisite relationships for these curriculum items are also sketched. The result is a topic “map” containing about one thousand topics. Several “knowledge base managers” are hired to create metadata for the existing Web based material, i.e. to use the topic authoring tool to create topics in the topic server’s knowledge base. First a rough associative mapping is created between all of the existing Web tutorials and the list of topics generated by the curriculum committee. This mapping is done with an off-the-shelf spreadsheet application (though a specialized tool for streamlining the task could be imagined). A given topic may be covered in several tutorials in different ways, and a given tutorial may cover many topics. The knowledge base managers know the domain of electronics well, but may not feel qualified (or may not have the time) to analyze each existing tutorial or Web page for the topics it addresses. Therefore a digital form is emailed to the authors of all of the Web-based content, asking them to indicate which topics (from a list predefined by the curriculum committee if possible) are addressed. The form also asks for other metadata, such as the teaching style and difficulty level of the tutorial (with values chosen from a pre-defined list from the topic ontology). It is expected that only about half of the tutorial authors will make the effort to complete and return these forms. Knowledge base managers make use of this information to complete the topic knowledge base, using the topic authoring tool.

**Topic authoring.** The topic authoring tool runs within a Web browser, and presents a form to be filled out to create a new topic. The form uses text entry boxes and pop-up menus to enter all of the data shown in the Example Topic and Request Section above to define a topic instance, including its name, description, relationship with other topics, parameters, attributes, and qualifiers. The tool automatically limits the author’s field values to satisfy the constraints defined in the IEEE-LTSC-1 topic ontology. When the form is completed the data is entered into the topic server’s knowledge base.

When an author creates a topic instance she might first browse the topic knowledge base (using the knowledge base browsing tool) to see if there is already a topic name (category) that corresponds to what she wants to teach and/or to check for inconsistencies between her topic name and other topics. If the topic category already exists in the knowledge base then the new topic will become another instance of that category; otherwise a new topic category will be created when the new topic instance is created. Alternatively, the authoring tool could constrain the topic categories to those already defined in the ontology (e.g. from a national educational standard).

**Learner-initiated tutorial searches.** A knowledge base filled in as indicated above does not include any “links” from one tutorial to another, but it does allow for learners who have a learning goal to search the knowledge base for a tutorial that addresses that goal. The system also has enough information to inform learners about prerequisites, and to present some of the metadata to them in a way that will help them select the best tutorial. For example, when a learner searches for a tutorial on “Ohm’s law” the server might return this message: “Ohm’s law can be learned from tutorial-A and tutorial-B. Tutorial-A uses a pedantic teaching strategy, has low difficulty, and takes 30 minutes to use on the average. Tutorial-B uses a learner-centered teaching strategy, and takes 45 minute to use on the average. Please select one.” In addition to supporting user-choice of several tutorials that match a pre-defined request, the server also offers a Web page that allows the student to search the topic knowledge base for arbitrary topics of interest.

**Authoring topic requests.** An authoring tool called the “tutorial request author” is provided that facilitates inserting a topic server request into a new or existing tutorial. These requests are what allow one tutorial to invoke another. The tool is another simple form that constrains input to comply with the descriptive vocabulary of IEEE-LTSC-1. Authors fill in the form to specify the topic to be taught, and other constraints such as the time length, the granularity, the learning level, the instructional style, etc. The tool provides authors with a hierarchical list of the topics already defined in the knowledge base, as well as a tool for browsing the contents of the knowledge base in detail. When they are done the tool creates a correctly formatted call to a small Java applet called “TopicRequest.class”, which the author ‘cuts’ and ‘pastes’ into an HTML tutorial (or, with minor modifications, into the code of any Web-based application). The Java applet handles the communication between the browser client and the topic server, to produce the behavior described below.

**The learner’s perspective.** We will assume that the system is now on line and students are registered and using it. When a new student registers (or is registered by administrative personnel) information about learning preferences and topics they already know about (perhaps as inferred from traditional courses they have taken) is entered. At some point in an HTML tutorial a student reads “You will need to understand Ohm’s Law before progressing further in this module. Press HERE if you want a tutorial on Ohm’s Law.” The “HERE” text is associated with a call to the TopicRequest.class applet that sends these parameters: topicName=OhmsLaw, difficulty=low. Once the student clicks on the HERE link the request is sent from the applet to the server and the server then returns. The student is shown a list of three tutorials that satisfy the request, in order of the closest match, as indicated above. The student chooses one and begins the Ohm’s Law tutorial. When it is completed, a message is given to the student orienting them to the fact that they are about to return to the original tutorial.

## 10. Other Web-based Curriculum Projects

This work is related to, yet distinct from, a number of other efforts to exploit the Web for educational uses. A large number of commercial and academic organizations are offering learning resource Web sites. The vast majority of these are search engines, information hubs, or electronic clearinghouses, for educational services (for example, see the University of Illinois Learning Resource Server<sup>18</sup>). These include catalogues of educational Web pages, software, and curriculum materials (such as project and activity descriptions), and some include electronic chat rooms or bulletin boards for students and educators that can be used to supplement traditional educational settings. Authoring and administration systems such as TopClass and WebCT are being used to manage Web based courses and student records. Some (see Chellappa, Barua and Whinston 1997; Osin and Lesgold, 1996) propose to completely re-engineer how education works by constructing distributed systems for mass customizing and delivering content on an individual level and managing all aspects of an educational program.

Most current efforts catalog educational resources at the course or Web site level of granularity. The most flexible customization of courseware will require locating and assembling educational resources at a smaller grain size. The Educational Object Economy<sup>19</sup> (introduced in Spohrer, Sumner and Buckingham Shum, 1998, this issue) is a large directory of freely available educational Java applets. The EOE's goal is to promote software reuse among communities of educational software authors. The EOE is one of the few educational resource sites to include reviewer's and comments about its contents. Also, it is one of the first educational resource sites to use the emerging educational metadata standards.

The model for distributed curriculum (MDC) relates to the above mentioned efforts in several ways.

1. It deals only with stand-alone tutorials (not Java applets or Web sites)
2. It allows for the reuse and interoperation of small grain size material, as well as large (i.e. tutorials dealing with a single concept can be integrated, as can entire courses)
3. It makes use of emerging educational metadata standards
4. Unlike any of the other projects, it includes a method for linking one tutorial to another at run time, and most significantly
5. It includes an automatic mechanism for retrieving and utilizing educational resources,

<sup>18</sup> *University of Illinois Learning Resource Server* <<http://lrs.ed.uiuc.edu/>>

<sup>19</sup> *Educational Object Economy, The EOE Foundation* <<http://www.eoe.org>>

whereas the other projects include standard search engines for finding resources, but it is left to the author to manually re-use the resource.

## **11. Research Questions and Future Work**

As mentioned in the introduction, this paper describes a model that has been only partially completed and has yet to be used and evaluated. As such, its contribution is primarily in helping to articulate some questions, goals, and issues in a new and rapidly changing field, and presenting one possible scenario toward these goals. The “answers” will only be found with more experience. Near term goals are to complete the implementation of MDC, and test it in the context of a single university course. (Possibly geology, as we are engaged in a project to support curiosity-driven exploration of information spaces, and are using geology as the first domain. See Murray, Condit and Haugsjaa, 1998). The major open questions for the model are listed below.

**Vocabulary uncertainty.** As has been mentioned in several contexts above, the degree to which terminological ambiguity will effect the serviceability of MDC is an open question awaiting trial use.

**Tutorial Gain Size.** We have evidence that the modularization of instructional content works well at two extremes of grain size, but little evidence for intermediate grain sizes. At the level of the academic “course,” a large grain sized unit, we have ample evidence that modules can be well defined, and flexibly and strategically sequenced to fit each learner’s needs (at least in many traditional subjects). A course in “linear algebra” taken at one university will suffice as the linear algebra prerequisite in another university, even though surely the courses at the two universities are not identical. At the other extreme, there is mounting evidence in the field of intelligent tutoring systems for the effectiveness of breaking subject matter into very small units, on the order of single procedural steps or rules, and generating instruction (explanations, activities, feedback, and user models) based on these small units (Anderson, 1990). Yet there is little evidence of the effectiveness or practicality of decomposing instructional content into intermediate sized units for the purpose of flexible and modular re-composition.

**Fuzzy Domains.** MDC is likely to work best in domains with highly structured or well-defined knowledge or objectives. These include traditional domains (those that have been taught and somewhat codified over the years) and quantitative domains (math, science and engineering). Domains in which it is hard to decompose the material into discrete topics, or in which there is little agreement as to the meaning of the topics, will be less amenable to MDC. MDC is applicable to all domains, but to the extent that the curriculum is less standard, the user community will have to be smaller and more cohesive to be able to manage the terminological

uncertainty. Also, it seems certain that the system will work better to the extent that topics are defined in terms of behavioral objectives. This is because it is difficult (if not impossible) to define what understanding a topic means without some reference to what students who have that knowledge can *do*. Osin and Lesgold also advocate for a topic-based curriculum structure (in the context of support project-based learning), and describe how a topic-to-project (or task) mapping facilitates more flexible, efficient, individualized learning.

**Limited Author Diligence and Metadata Expense.** Our limited experience in managing metadata for educational objects (see Murray and Haimovitz, 1998) indicates that a majority of authors of tutorials will be reluctant to author metadata information for their tutorials. The incentives are not high enough yet. This means that third-party analysis of existing tutorials will be needed for most of the metadata, which is time consuming and expensive. Analyzing a tutorial to identify the concepts it addresses, its interaction quality, educational level, difficulty, etc. will be much more time consuming than, say analyzing a Web page to enough depth to catalogue it in a Yahoo data base.

**Infrastructure Costs.** For real world robust use of MDC in industrial or academic settings, a large number of other services need to be in place, including registration and rostering, evaluation, and report generation. Validation/accreditation, assessment/certification; security/authenticity; and intellectual property rights and licensing methods need to be worked out (see Hiltz and Wellman, 1997). Much of this software is under development for virtual universities. Hamalainen, Whinston and Vishik (1996) advocates for the emergence of electronic markets for learning and “education brokers” who will locate, organize, and even package and deliver Web-based educational material for customers with specific education or training needs. MDC seems quite applicable to such contexts.

## **12. Conclusion: From Content-Centered to Knowledge-Centered Curriculum Management**

Several factors contribute to the growing need for distributed education and training and just-in-time, on-demand education. These factors include the rapidly growing and changing nature of the information and skills; the fact that organizations are becoming more team-based, individualized, networked, and geographically dispersed; and the subsequent increasing cost of traditional instructional methods (Hamalainen, *et al.* 1996).

Though there are a number of efforts to catalog, share, distribute, and/or re-use educational resources on the word wide Web (discussed above), all of these are essentially manual systems. Search tools help locate the resource, but reusing the resource is not automated. Our model for distributed curriculum (MCD) uses a topic server architecture to allow one Web-based tutorial

to include a specification for another tutorial, a “virtual tutorial”, where the best match will automatically be found at run time. This allows a designer of instructional material to re-use other Web-based tutorials without having to know what tutorials will exist at run time, i.e. without having to search and analyze material on the Web. The scheme is intended to allow tutorials to be “encapsulated” by topic objects, allowing the re-use of existing tutorials without having to alter them.

MDC (an incompletely implemented model, which has yet to be tested) incorporates the following major features:

1. A knowledge base of topic objects that refer to knowledge or learning objectives, where each topic points to one or more Web-based tutorials that can teach (or facilitate the learning of) that topic;
2. A vocabulary (or ontology) of metadata tags for describing the pedagogically relevant properties of the topics and tutorials, based on emerging metadata standards for educational objects;
3. A Web-based topic server that contains the topic knowledge base and can process requests for tutorials that teach particular topics, and/or have particular pedagogical characteristics (such as granularity or difficulty);
4. An method for inserting tutorial requests into existing Web-based tutorials and returning either the tutorial URL or a list of URLs for tutorials that match the request
5. An interface that allows learners to request tutorials and select among a set of tutorials that partially match a request;
6. Authoring tools for modifying and browsing the topic knowledge base and for creating tutorial requests embedded in existing tutorials;
7. A simple learner model for tracking what tutorial users have visited and preventing repetition of material.

MDC proposes a shift from tutorial-centered (or content-centered) to topic-centered (or “knowledge-centered”) management of learning sessions. Table 1 summarizes the main differences between the approaches.

<b>Tutorial-centered</b>	<b>Topic-centered</b>
The learner's performance scores are kept on a per tutorial basis (an "overlay" on tutorials).	A value is given to each topic summarizing the learner's experience with or mastery of that topic (an "overlay" on knowledge: concepts, skills, etc).
Prerequisite and other links are defined in a space of tutorials.	Prerequisite and other links are defined in a space of topics.
Easier to author and manage.	Harder to author and manage
Less knowledge "depth," less conceptual fidelity.	More knowledge "depth," More conceptual fidelity.

Table 1: *Tutorial-centered vs. topic centered curriculum representations*

The topic-centered approach is more flexible. Instructional designers can represent learning, prerequisites, etc. in terms of the knowledge or skills involved, and not have to specify or even be aware of what tutorial will be used. However, the extra level of abstraction and uncertainty involved in the topic-centered approach results in more complexity. These are open questions:

1. Can teachers and instructional designers really work at this abstract level?
2. Is it possible to cleanly decompose a tutorial into the topics it teaches?
3. Is it possible to map out a curriculum of knowledge and skills in such a way that alternative methods of learning (alternative tutorials) can be used to learn each topic?

These questions are matters of degree. More experience and research is needed to learn what contexts and domains are best suited to the method. I have described several methods to mitigate these problems: using MDC in controlled contexts (such as a consortia of organizations), making use of emerging metadata standards, and including student choice in the tutorial retrieval process, so as not to overestimate the "intelligence" of the topic server. Future work will inform us about the strengths and limitations of the method in actual use, and will provide information for improving the vocabulary of pedagogical characteristics that MDC uses.

## References

- Anderson, J.R. (1990) *Analysis of Student Performance with the LISP Tutor*. In Frederiksen, N., Glaser, R., Lesgold, A.M. and Shafto, M. (Eds.), *Diagnostic Monitoring of Skill and Knowledge Acquisition*. Hillsdale, NJ: Lawrence Erlbaum.
- Brusilovsky, P, Schwartz, E. and Weber, G. (1996). *ELM -ART: An Intelligent Tutoring System on the Work Wide Web*. In Proceedings of ITS-96, Frasson, Gautheir, Lesgold (Eds.), Springer: Berlin, 1996. pp. 261-269.
- Chellappa, R., Barua, A. and Whinston, A.B. (1997). *An Electronic Infrastructure for a Virtual University*. *Communications of the ACM*, 40 (9), 56-58.
- Fikes, R., Cutkosky, M., Gruber, T. and Van Ballen, J. (1991). *Knowledge Sharing Technology: Project Overview*. Technical Report 91-71, Knowledge Systems Laboratory, Stanford University, CA.
- Finn, T, Fritzon, R. and McKay, D. (1992). *An Overview of KQML: A Knowledge Query and Manipulation Language*. Technical Report, Department of Computer Science, University of Maryland Baltimore County.
- Gruber, T. (1993). *A Translation Approach to Portable Ontology Specifications*. *Knowledge Acquisition*, 5 (2), 199-220.
- Gruber, T. (1994). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Guarino and Poli (Eds.). Kluwer Academic Publishers.
- Guha, R. V. (1996). *Meta-Content Format*. Apple Computer Working Paper <<http://mcf.research.apple.com/projectx/mcf.html>> .
- Hamalainen, M., Whinston, A.B. and Vishik, S., (1996). *Electronic Markets for Learning: Education Brokers on the Internet*. *Communications of the ACM*, 39 (6), 51-58.
- Hiltz S.R. and Wellman, B. (1997). *Asynchronous Learning Networks as a Virtual Classroom*. *Communications of the ACM*, 40 (9), 44-49.

Kiyama, M., Ishiuchi, S., Ikeda, K., Tsujimoto, M. and Fukuhara, Y. (1997). *Authoring Methods for the Web-Based Intelligent CAI System CALAT and its Application to Telecommunications Service*. In the Proceedings of AAAI-97: Conference of the American Association for Artificial Intelligence, Providence, RI.

Major, N.P. and Reichgelt, H. (1992). *COCA - A Shell for Intelligent Tutoring Systems*. In Frasson, C., Gauthier, G. and McCalla, G.I. (Eds.) Proceedings. of the International Conference on Intelligent Tutoring Systems (ITS 92). Springer Verlag, Berlin.

Merrill, M.D. (1983). *Component Display Theory*. In Instructional-Design Theories and Models: An Overview of their Current Status, pp. 279-333. C.M. Reigeluth. (Ed), Lawrence Erlbaum Associates, London.

Mizoguchi, R., Sinitsa, K. and Ikeda, M. (1996). *Task Ontology Design for Intelligent Educational/Training Systems*. Position Paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal, June, 1996.

Murray, T. (1996a). *Special Purpose Ontologies and the Representation of Pedagogical Knowledge*. Proceedings of the International Conference on the Learning Sciences, 1996. AACE, Charlottesville, VA.

Murray, T. (1996b). *Toward a Conceptual Vocabulary for Intelligent Tutoring Systems*. Working Paper, Computer Science Department, University of Massachusetts <<http://www.cs.umass.edu/~tmurray/papers.html>>.

Murray, T. (1998). *Authoring Knowledge Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design*. Journal of the Learning Sciences, 7 (1), 5-64.

Murray, T. and Haimovitz, R. (1998). Lessons Learned From Third-Party Authoring of Metadata for Educational Objects. Working Paper, University of Massachusetts (available from the authors).

Murray, T., Condit, C. and Haugsjaa, E. (1998). *MetaLinks: A Preliminary Framework for Concept-Based Adaptive Hypermedia*. In Workshop Proceedings for ITS-98 Workshop on WWW-Based Tutoring, August, 1998, San Antonio, TX.

Musen, M.A. (1992). *Dimensions of Knowledge Sharing and Reuse*. Computers and Biomedical Research, 25, 435-476.

Osin, L. and Lesgold, A. (1996). *A Proposal for the Reengineering of the Educational System*. In Proceedings of the International Conference on the Learning Sciences, 1996. AACE, Charlottesville, VA.

Paepcke, A., Chang, C.K., Garcia-Molina, H. and Winograd, T. (1998). *Interoperability for Digital Libraries Worldwide*. In Communications of the ACM, 41 (4), 33-43.

Ritter and Suthers (1997). *Technical Standards for Education*. Working Paper, Educational Object Economy site, EOE Foundation <<http://www.eoe.org>>

Spohrer, J., Sumner, T. and Buckingham Shum, S. (1998). *Educational Authoring Tools and the Educational Object Economy: Introduction to this Special Issue from the East/West Group*. Journal of Interactive Media in Education, 98 (10). <<http://www-jime.open.ac.uk/98/10>>

Stern, M.K., Woolf, B.P. and Kurose, J.F. (1997). *Intelligence on the Web?* Proceedings of the 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August, 1997.

Suthers, D. (1995). *Towards an Interlingua for Information Objects*. Position Paper, Basic Research Symposium, ACM CHI'95: Conference on Human-Computer Interaction, Denver, Colorado.

Swartout, B., Patil, R., Knight, K. and Russ, T. (1996). *Toward Distributed Use of Large Scale Ontologies*. In B.R. Gaines and M. Musen (Eds.) Proceedings of 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada. <[http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff\\_96\\_final\\_2.html](http://ksi.cpsc.ucalgary.ca/KAW/KAW96/swartout/Banff_96_final_2.html)>

Van Marcke, K. (1992). *Instructional Expertise*. In Frasson, C., Gauthier, G. and McCalla, G.I. (Eds.) Proceedings of the International Conference on Intelligent Tutoring Systems (ITS 92). Springer Verlag, Berlin.

Valente, A. and Breuker, J. (1996). *Towards Principled Core Ontologies*. Proceedings of 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada. <<http://ksi.cpsc.ucalgary.ca/KAW/KAW96/valente/doc.html>>

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.