

Reusable Multimedia Content in Web-Based Learning Systems

Abdulmotaieb El Saddik
Darmstadt University of Technology, Germany

Stefan Fischer
Mobile Video Communications, Germany

Ralf Steinmetz
German National Research Center for Information Technology

We discuss reusability aspects of interactive multimedia content in Web-based learning systems. In contrast to existing approaches, we extend a component-based architecture to build interactive multimedia visualizations by using metadata for reusability issues. The experiment we conducted shows how to reuse the same visualization in different learning contexts.

The primary motivation for using multimedia technologies in education is the belief that they will support superior forms of learning. Advances in cognitive psychology have inculcated our understanding of the nature of skilled intellectual performance and provided a basis for designing wholesome learning environments. There's now a widespread agreement among teachers, educators, and psychologists that we acquire advanced comprehension, reasoning, composition, and experimentation skills not only by the transmission of facts but also through interaction with content. Albert Bork, professor of physics at the University of California, Irvine, and strong computer-interaction advocate, describes interactive learning as "the most valuable aspect of the computer in education [...] with students constantly cast as participants in the process rather than spectators."¹ His work concentrates on using prim-

itive computer graphics to visualize physics concepts, but he also emphasizes that "there is not a single way but a whole variety of ways" to teach with computers.

Often, instructors don't use educational software because what's available is simply irrelevant to their curricular goals. Most of the more sophisticated, inventive pieces of instructional software deal with only a narrow slice of curriculum or with fairly trivial material. At the same time, instructors report intense pressure to cover an unrealistically large amount of required material.

One of the key problems in developing educational software systems in general and interactive instructional visualizations in particular is planning for change. (See the sidebar "Limitations of Instructional Software" for more information.) Educational systems must be flexible and easy to adapt to new, changing user requirements. The state-of-the-art way of dealing with open requirements is to build systems out of reusable components conforming to a plug-in architecture.² We can then change or extend the functionality of such systems by substituting or plugging in new components. Although the component-based solutions developed to date are useful, they're inadequate for developers and instructors building component-based interactive learning environments in which the components must respond to the meaning of the content as well as its form and presentation.

To address these concerns, this article introduces a new metadata classification scheme and an extension to IEEE Learning Object Metadata. We apply component-based solutions and metadata resources to interactive learning environments, developing easily adaptable and reusable visualizations. (For information about our starting point and the motivation for our work, see the sidebar "Multibook" on p. 32 for an overview of the Multibook project.)

Metadata

Metadata are data about data, descriptive information about resources for the purpose of finding, managing, and using them more effectively. We use this system of labels to describe a resource or object's characteristics and its objectives.

The starting point for our work is the existing technologies, standards, and ongoing initiatives in the multimedia educational metadata area. The Dublin Core³ Metadata Element Set, Educom's Instructional Management System, the Alliance of

Remote Instructional Authoring and Distribution Networks for Europe, and IEEE's Learning Object Metadata Working Group 12 (IEEE-LOM) are the most important initiatives dealing with metadata for computerized learning purposes. These initiatives closely relate to the Resource Description Framework (RDF), the Warwick Framework,⁴ and other World Wide Web Consortium activities. The purpose of metadata is to provide a common means of describing things (electronically) so that we can define, search, and find learning objects (however they're defined).

Learning content is only one area of metadata application. Researchers are actively developing metadata in all aspects of Web-based content and commerce.⁵ Today, the Internet abounds with resources. Looking for a specific topic or resource, users often find hundreds or thousands of hits and most don't meet their requirements. According to the IEEE-LOM, the advantages of using metadata include

- summarizing the data's content or meaning;
- letting users search for data;
- letting users determine if the data is what they want;
- preventing some users, such as children, from accessing data;
- retrieving and using a copy of the data; and
- instructing users how to interpret the data such as format, encoding, and encryption.

For content providers or publishers, metadata eases the discovery of and access to their resources so they can reuse them. Ensuring that users can locate resources should have a high priority.

For the person searching for material, metadata is helpful. It optimizes searches by narrowing the result lists to applicable resources. Those located will always be presented with minimum information such as creator, subject, type, format, and identifier. The metadata provider must enter this kind of information. If the resource meets the searchers' requirements, the resource's location will tell them where to obtain it.

Developers can store metadata separate from or together with the resource. Metadata on the Internet, for instance, must be machine readable and machine understandable. For example, meta-

Limitations of Instructional Software

Instructional software today is locally effective but globally fragmentary. Hence, to date, it has had limited impact on systematic curricular reforms and fails to meet large-scale needs for reuse of interactive content. For example, it's awkward to combine interactive visualization modules that are each valuable in their own niche and theoretically complementary in ensemble. Users, for example, can't connect an animation of a video decoder to that of a network and study the resulting effects.¹

Ongoing research in the instructional-visualization area sheds insight into factors that contribute to the design of effective visualization systems² It suggests that attempts over the last two decades to teach algorithm behavior with visualizations were unsatisfactory, not because of a flaw with the animation and visualization techniques, but perhaps due to the approaches used to convey the animations.

Today several standards efforts specifically target advanced educational technologies (see <http://ariadne.unil.ch> and <http://www.imsproject.org>), and researchers have built repositories for educational object components (see <http://www.eoe.org> and <http://www.geminfo.org>). These efforts gain leverage from the rise of interactive Web technology and its associated emphasis on standards-based interoperability. Emerging solutions for component-based systems include development frameworks (JavaBeans and ActiveX), the shared communication protocols (Remote Method Invocation and Corba), markup languages (Hypertext Markup Language and Extensible Markup Language), and metadata formats³ (see <http://www.imsproject.org>).

References

1. A. El Saddik et al., "A Component-Based Construction Kit for Algorithmic Visualizations," *Proc. IDPT*, Springer-Verlag, New York, 2000.
2. S. Hansen, D. Schrimpscher, and N. Narayanan, "Learning Algorithms by Visualization: A Novel Approach Using Animation-Embedded Hypermedia," *Proc. Third Int'l Conf. Learning Sciences*, Association for the Advancement of Computing in Education, 1998, pp. 125-130.
3. *Provisional Report of the Dublin Core Supplement Working Group*, 2000, <http://dublincore.org/documents>.

data separate from the resource can be in a database, and metadata together with the resource can be at the top of a document.

When we studied and considered metadata standards, an important disadvantage became obvious. We can properly describe static resources, such as images or text documents, with metadata but appropriate descriptions of dynamic resources, for example animations, remain feasible only to a limited extent. The reason is dynamic multimedia objects can process input parameters, generate output parameters, and work internally with data that traditional metadata schemes can't describe. Therefore, changes in the granularity of metadata prove necessary to match users' learning goals and to reuse dynamic multimedia content in different contexts.

Multibook

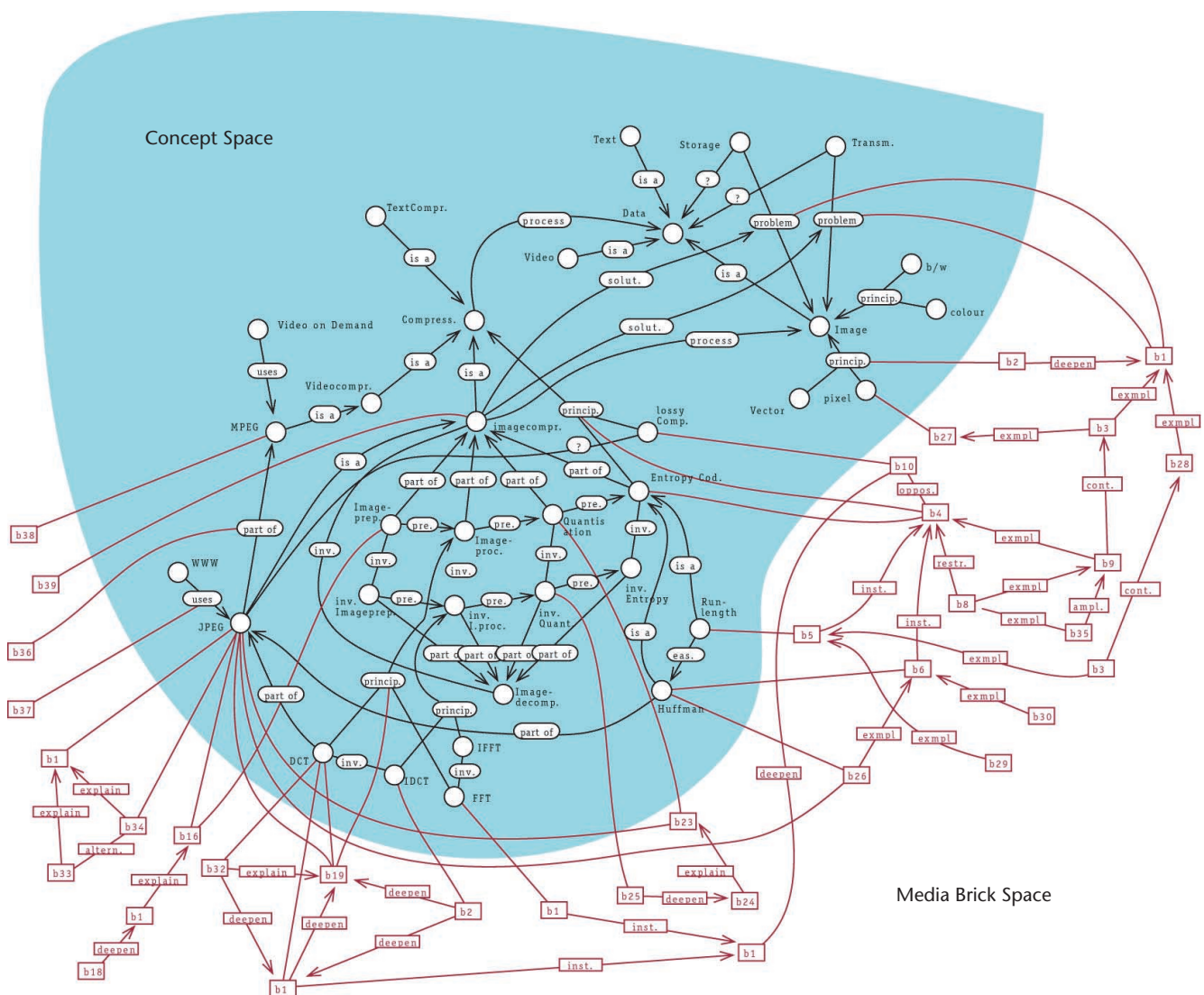
The Technical University of Darmstadt and the University of Hagen (see <http://www.multibook.de/English/english.html>) developed the Multibook. It's a Web-based adaptive hypermedia learning system for multimedia and communication technology. Multibook aims to offer different lessons for different users, either by storing a huge number of compiled lessons—the disadvantage of this approach isn't the amount of storage but the static character of such lessons—or by dynamically generating the lessons individually for each user. Because the dynamic composition also facilitates exchanging or modifying information, Multibook follows the second approach.

Multibook uses four dimensions for the user profile. Initially, we fill the profile with the users'

demands and preferences. While the users work with Multibook, the system keeps track of what information they have already seen and learned, what additional material they request, their test results, and so on.

Multibook's knowledge base consists of two separated knowledge spaces (see Figure A).¹ The Concept Space contains a networked model of learning topics and uses well-known knowledge-management approaches. The knowledge topics are interconnected via semantic relations. The media bricks stored in the system's Media Brick Space are atomic information units of various multimedia formats. These units are interconnected via rhetoric relations.² Each media brick is described by the author using the IEEE's Learning Objects

Figure A. Part of the Multibook network for the multimedia book.



Metadata (IEEE-LOM) scheme—we refer to media bricks as learning objects. Although the information spaces are separate, each learning object can have a relation to one or more related topics. Multibook generates adaptive lessons by separating these spaces, because for each topic, a set of media bricks (texts in different granularity, animations, video, and so forth) is available. The selection of media bricks is then determined by each user’s preference.

Figure B shows Multibook’s architecture, which is similar to the one proposed by the IEEE Learning Technology Standards Committee (LTSC) LOM group.³

Considering the way an author writes a document, we can specify the following tasks:

- an author acquires background knowledge,
- an author creates an outline for a document, and
- an author fills the outline with content.

Different spaces in Multibook model these steps. The Concept Space contains an ontology of keywords to create a lesson’s outline. After ordering the outline (the equivalent of creating a table of contents), the actual content (text, images, audio, video, and animation) is filled by the lesson generator agent into the outline using elements of the second space, the Media Brick Space. A general abstraction of Multibook is that it must employ different relations within the Concept Space and the Media Brick Space to model the different goals that both spaces have. There are objects (concepts and media bricks), relations (semantic relations in the Concept Space, and rhetorical–didactic relations in the Media Brick Space), and attributes of the media bricks. We use the facts in the Concept Space as an index to the original learning material—an approach much closer to publishing practice than

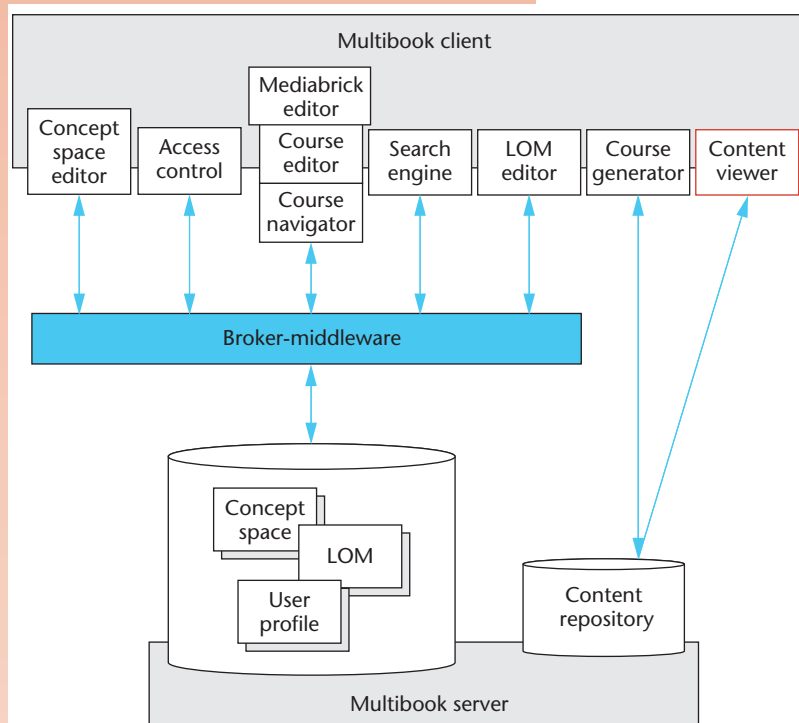


Figure B. Multibook architecture.

to IT or expert systems—so they won’t mirror each other that’s made in the media bricks.

References

1. C. Seeberg et al., “From the User’s Needs to Adaptive Documents,” *Proc. IDPT*, CD-ROM, 2000.
2. W. Mann and S. Thomson, *Rhetorical Structure Theory: A Theory of Text Organization*, tech. report RS-87-190, Information Science Inst., USC ISI, 1987.
3. A. Faatz et al., “Multimedia und Wissen: Unser Weg zu einem produktiven Umgang mit Wissensdurst,” (Multimedia and Knowledge: Out Way to a Productive Handling of Knowledge), *Thema Forschung* (Research Topics), vol. 2, Nov. 2000, pp. 26-33.

Metadata specification for interactive learning objects

Metadata descriptors are fixed because their granularity remains as the original metadata author defined it. Such metadata can’t adequately describe interactive visualizations. Moreover, it can’t influence the multimedia content because metadata usually contain universal and widely applicable object descriptions.

Figure 1 (next page) illustrates the traditional

way of tagging a learning object, using the IEEE-LOM scheme. In this tagging process, many descriptions can exist simultaneously in order to describe specific content. To describe n resources, we might need m descriptions, where $m > n$. An ideal situation has exactly one description for every content ($n = m$).

The categories of the IEEE-LOM base scheme can cover diverse meta-information about the instructional visualizations. We can use the IEEE-LOM to

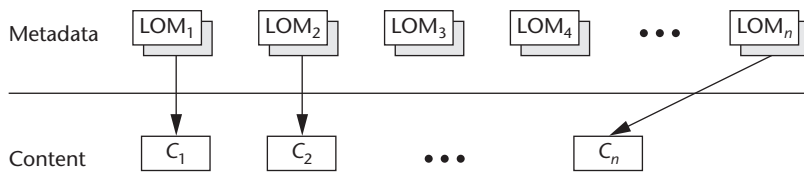


Figure 1. *Ideal traditional metadata tagging process.*

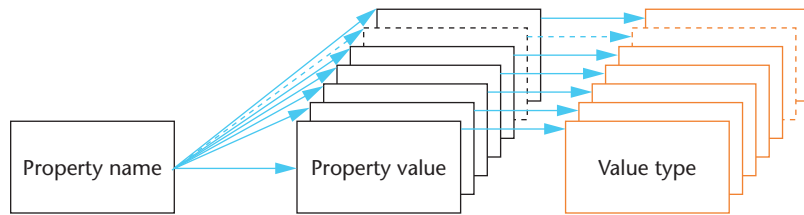


Figure 2. *Generic scheme of dynamic metadata.*

search and navigate as long as we use static learning objects. However, we can only exploit the particular potential of interactive visualizations (their flexibility and adaptability) to a limited extent. For example, we can use some interactive visualizations to illustrate an algorithm's different scenarios or parts, depending on the input parameters. We can reuse the same learning object in a different learning context, according to the way it's configured by parameters. But how do we tag an instructional visualization illustrating several behaviors?

An intuitive and simple answer is that for us to use an interactive visualization in n arbitrary different scenarios, it should be parameterized, stored (with the appropriate configuration), and tagged (assuming the ideal tagging scenario) n -different times.

Another problem with such a situation is that the number of scenarios an interactive visualization can illustrate depends on the number of attributes (parameters) a visualization component has and the number of values this parameter is allowed to have. If for instance an interactive visualization contains three attributes, each of them can have five different values. This visualization can, therefore, illustrate 15 different situations. Thus, following the standard metadata tagging scheme, the person in charge of the tagging process must describe the visualization 15 different times to use it in all its possible scenarios. This is unacceptable and unrealistic, even when using templates because IEEE-LOM, for example, has approximately 65 fields.

Automatically generating a lesson is another problem we encounter with today's metadata. IEEE-

LOM, among others, doesn't provide enough granularity. For instance, it doesn't specify a learning resource's physical size in pixels or dots—it gives size in bytes. To integrate an interactive visualization into a dynamically generated lesson, we require other elements such as the height and width.

Recognizing the high cost of tagging and using instructional visualizations doesn't help us unless we can reduce it. We can parameterize interactive visualizations either offline or online. Traditional metadata assume that we are parameterizing interactive visualizations offline. For online customization, we propose using dynamic metadata to extend the static IEEE-LOM.

Extension of IEEE-LOM

Initially, we wrote informal textual or Unified Modeling Language (UML) descriptions capturing the important information about a representative sample of interactive visualizations that we wanted to describe and use in different contexts. After reviewing these descriptions using the IEEE-LOM Draft 4 and Draft 5, we went through them and identified IEEE-LOM elements in which the information expressed could be captured. Where we failed to find LOM elements for an item, we extended the IEEE-LOM either by expanding the vocabulary of an existing element or by creating an entirely new element under a new category. Where we needed new elements, we searched other repositories to find metadata that we could use, including the Gateway to Educational Materials (see <http://www.geminfo.org>) and the Advanced Distributed Learning Initiative (see <http://www.adlnet.org>).

Users can understand dynamic metadata as an extension of IEEE-LOM. This categorization groups the information to define the behavior of dynamic learning objects or instructional visualizations. The scheme of dynamic metadata follows the generic format **<property, values, value type>**. Figure 2 illustrates this scheme.

Flexible visualizations

A visualization must be flexible enough to confront and address changing user requirements and knowledge. It should also be versatile and usable in a variety of contexts. Although an off-the-shelf visualization often won't fit a particular user's needs, we can tailor it for a better fit. Using dynamic metadata, educators can convert an algorithm that a developer implemented to a series of animation sequences by mapping algorithm variables, specifying animation actions, and associat-

ing execution points in the algorithmic chain to perform the desired animation.

Educators, therefore, become designers of the visualization. They can customize the learning object to visualize a desired behavior that's appropriate for the course they are teaching. Thus, they use and reuse already developed instructional objects accompanied with dynamic metadata. Dynamic metadata let educators

- tag a resource once and use it in different scenarios,
- tailor interactive visualizations according to user needs, and
- integrate interactive visualizations in an appropriate learning context.

Component-based development of instructional visualizations

Components are autonomous, reusable software entities that work as primitive buildings blocks whose behavior users can program. They are designed to be combined with other components in user-defined configurations and behave as composite constructs. Component-based design (CBD) methodologies are based on many of the same principles as object-oriented design and use the same diagrams and notations as many object-oriented methodologies.

Developers design component-based software systems at two levels. In the specification level, a developer understands and describes a problem. This process includes analysis and design, which results in a potential solution for a software application. Developers can express the solution in graphics or text using a notation such as UML. An implementation level is the realization of the specified solution with a programming language and other development tools. This approach means that the products of the development process will include software as well as diagrams, models, and other specifications.

The steps that are traditionally associated with software development—analysis, design, and implementation—still exist in the CBD method. We categorize them under the headings of either specification or realization. The design step actually crosses the boundary between specification and realization because some tasks design the specification and some design the implementation details. CBD is an iterative process. As we move from the specification to

**We must strike a balance
between flexibility and
complexity, depending on the
level of abstraction, likelihood
of change, and so forth.**

the realization stage, and as the design and development moves from the analysis to the implementation phases, we can go back to refine earlier phases.⁶

Furthermore, specification components don't depend on realization technology. This provides a developer with the flexibility to use a given specification component in different applications or environments. For example, we can use a UML specification in any object-oriented development environment. A developer might use C++, Java, and Visual Basic implementations. A component's value, therefore, isn't exclusive to its software implementation but to its specification.

Various factors that help define the range of component granularities drive component design decisions. Generally, components should be larger rather than smaller. However, larger components by nature have more complex interfaces and represent more opportunity to be affected by change. The larger the component, the less flexible the system's structure. We must strike a balance between flexibility and complexity, depending on the level of abstraction, likelihood of change, and so forth.⁷ The principles of cohesion and coupling are the factors; minimizing the coupling of the system tends to work against good cohesion.

Components can be as large as whole applications such as a PDF viewer. But examples of small components include many of today's available graphical user interface components, also called widgets, that are available from many development organizations. They are often implemented as JavaBeans or ActiveX components. Another important aspect of components is specificity. The more closely a component matches the design, the less modification it requires. Naturally, the number of components increases as they become more specific.

Large components, also called coarse-grained components, for interactive visualization illustrate

Table 1. Ethernet applet components.

Component	Symbol	Description
Host		Represents a computer that accesses the shared Ethernet channel.
Arrow		Represents the data flow within a host. A down arrow represents a sender, and an up arrow represents a receiver.
Bus element		Represents the Ethernet channel's smallest entity.
Signal		Represents the Ethernet channel's state. Blue indicates a normal transmission, red a collision, and yellow padding.
Bus segment		Each bus segment consists of several bus elements and has four I/O possibilities: top, bottom, left, and right. The signal traverses a bus segment.

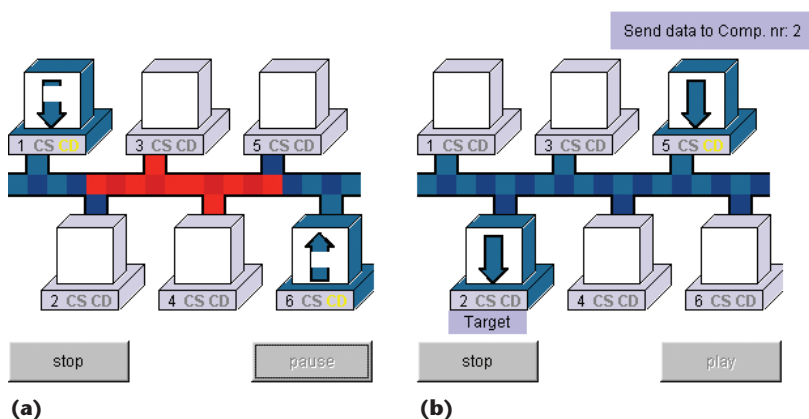


Figure 3. Parametrizations of dynamic media bricks.

only some of the intermediate computations that take place within an algorithm.⁸ Coarse-grained components are appropriate for certain concepts, mainly for those courses where instructors teach self-contained algorithms and data structures. However, many concepts and topics in multimedia communication are combinations of small concepts that provide parts of a theoretical framework for larger algorithms. The visualization of JPEG or MPEG⁹ serves as a good example. Even though both compression schemes use the discrete cosine transform (DCT) and the Huffman encoding, we can't reuse a component of a JPEG animation in most cases to visualize a step of the MPEG-compression process if it's coarse-grained. Coarse-grained animations are useful in demonstrating the final concept but are hard to use in teaching parts of a concept.

The best granularity of a module, therefore, strongly correlates with the domain being addressed and varies widely between concepts.

Putting it all together: Visualization of the Ethernet

We developed a lesson explaining the IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Ethernet protocol¹⁰ to demonstrate the reusability issue during the development stage and the opportunities that parameterizing an animation can offer. (Visit <http://www.multibook.de> for a German version of the lesson.) The lesson provides an interactive environment that elicits active student participation, using a carefully orchestrated presentation of information in various media, such as text, visualization, static diagrams, and interactive simulations with appropriate temporal, spatial, and hyperlink connections.

Decomposition

Ethernet technology predates the IEEE's Local Area Network standards committee, so the first Ethernet standard was developed by a vendor consortium made up of Digital Equipment, Intel, and Xerox. This was the first open standard for LAN technology ever published. The first Ethernet-like IEEE standard was published in 1985 and formally called the *IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer* specifications. The Ethernet system consists of three basic elements:

- the physical media used to carry Ethernet signals between computers,
- a set of media access control rules embedded in each Ethernet interface that let multiple computers access the shared Ethernet channel, and
- an Ethernet packet or frame that consists of a standardized set of fields used to carry data over the system.

The Ethernet uses a bus topology, a linear networking architecture that generally uses one or more pieces of cable (bus element) to form a single line (bus segment), or bus. The signals sent by one station (host) extend the length of this cable to be heard by other stations. Taking this into account, together with the explanations of the previous sections of this article, and doing a first-level decomposition of the IEEE 802.3 protocol, we identified five components. We developed each of these components (see Table 1) as a JavaBean. These components are the basic elements we used to implement an applet visualizing the diverse functionality of Ethernet.

Parade lesson

We implemented a lesson in HTML 4.0 that contains visualizations we developed according to the JavaBeans component-based framework enhanced with the dynamic metadata set El Saddik et al. describe.¹¹ The lesson consists of 15 pages.

In the lesson, we first explain the Ethernet's functionality. After that, the student answers the question, Which problems must be faced in a bus topology? We provide a set of answers and use the same animation to explain why the answers are correct or wrong. We explain the difference between the answers with different parameterizations of the same animation. We stored these parameterizations as dynamic metadata for interactive visualizations. Figure 3a shows an example. The system asks, Why is the protocol complex? One possible answer is, "The protocol is complex because messages can't be sent to a specific computer. They can only be sent to several computers at once."

If we use dynamic metadata, we can use another scenario of the same animation to provide an answer to the question, How can the collision problem be solved in Ethernet? A possible answer is, "Collisions can't be avoided. If a collision is detected the transmission has to be repeated." If a collision occurs, the transmitting station recognizes the interference on the network and transmits a bit sequence called *jam*. The jam ensures that the other transmitting station recognizes that a collision has occurred. After a random delay, the stations attempt to retransmit the information and the process begins again.

Figure 3b illustrates a parameterized version of the animation in Figure 3a. Because we use dynamic metadata, we can reuse the same animation in a different context. Figure 4 shows a screen shot of the lesson.

The lesson ends with the option of having an interactive simulation of the Ethernet algorithm. The simulation consists of three levels. In the first level, users can choose only one out of two scenarios (normal operation and a collision problem). In the second, they can choose more scenarios. In the third, students can build their own scenarios.

User evaluation

Thirty-two graduate students from Darmstadt University of Technology participated in our experiment. They all interacted with the lesson over the Web at <http://www.multibook.de> or on a CD-ROM that included the Multimedia Book written by Steinmetz.⁹ We wrote the questions in

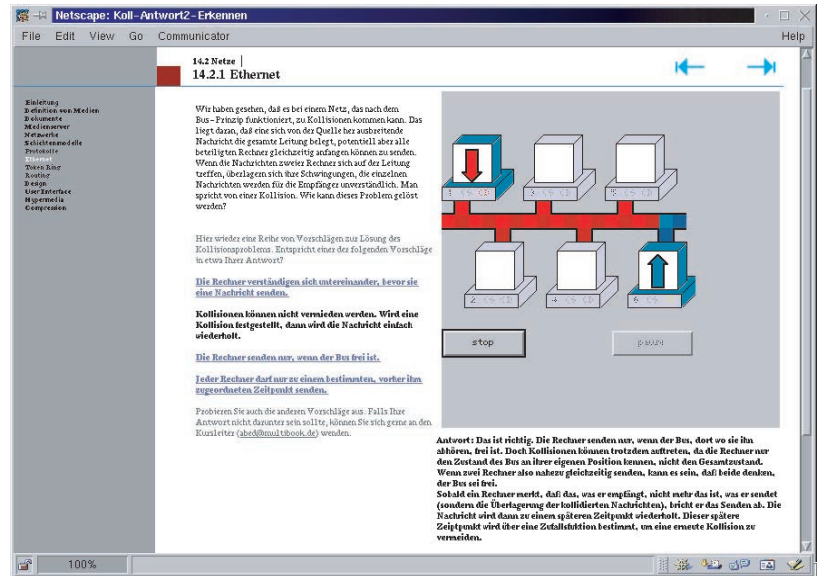


Figure 4. Screen shot of the example lesson.

coordination with cognitive scientists from the Justus-Liebig University of Giessen.

One goal of our study was to show the benefit of developing a component-based visualization accompanied with metadata. We developed the lesson to prove that by using dynamic metadata, we can adapt visualizations to a situation without harming usability. In our sample lesson, the students reused the Ethernet visualization eight times with different parameters, illustrating different scenarios and phenomena. We ensured interoperability and platform independence by using HTML and Java.

Eighty-two percent of the students said that the visualizations didn't bore them. Therefore, the applets' similarity doesn't seem to be a disadvantage. Six students felt that the visualizations were monotonous and another didn't enjoy viewing them at all. Because only a small number of students criticized particularly the visualizations' similarity, we believe usability didn't suffer when we customized and reused the applet.

Conclusions

After assessing the users' responses, we found that the strength of instructional visualization integrated in the appropriate learning environment seems to be in the autonomous and continuous knowledge-acquisition and refresh processes.

Our experience suggests that the visualizations our system offers can provide an environment in which an educator without conventional programming skills can build a useful, interactive, and visual algorithm relevant to a particular task. We'll

continue to extend the systems, particularly by increasing the number of available visualization units. Currently, we're using our architecture to develop other examples of teaching animations—such as explaining multimedia scheduling algorithms—for multimedia and communications courses in the Department of Electrical Engineering and Information Technology at the Darmstadt University of Technology. **MM**

References

1. R. Taylor, *The Computer in the School: Tutor, Tool, Tutee*, tech. report ASIN: 0807726117, Teachers College Press, New York, 1980.
2. A. El Saddik, S. Fischer, and R. Steinmetz, "ITBeanKit: An Educational Middleware Framework for Bridging Software Technology and Education," *Proc. EdMedia 2000*, Springer-Verlag, New York, 2000.
3. *Dublin Core Metadata Element Set*, version 1.0, <http://dublincore.org/documents>, 2000.
4. C. Lagoze, "The Warwick Framework, A Container Architecture for Diverse Sets of Metadata," *D-Lib Magazine*, vol. 2, no. 7/8, July/Aug. 1996, <http://www.dlib.org/dlib/july96/lagoze/07lagoze.html>.
5. O. Lassila, "Web Metadata: A Matter of Semantics," *IEEE Internet Computing*, vol. 2, no. 4, July/Aug. 1998, pp. 30-37.
6. M. Sparling, "Lessons Learned Through Six Years of Component-Based Development," *C. ACM*, vol. 43, no. 10, Oct. 2000, pp. 47-53.
7. J. Hopkins, "Component Primer," *C. ACM*, vol. 43, no. 10, Oct. 2000, pp. 27-30.
8. D.L. Gould, R.M. Simpson, and A. van Dam, "Granularity in the Design of Interactive Illustrations," *Proc. ACM SIGCSE*, ACM Press, New York, 1999, pp. 306-310.
9. R. Steinmetz and K. Nahrstedt, *Multimedia Computing, Communications, and Applications*, Prentice Hall, Upper Saddle River, N.J., 1995.
10. *ANSI/IEEE Standard IEEE-802.3: Ethernet*, IEEE, Piscataway, N.J., 2000.
11. A. El Saddik et al., "Metadata for Smart Multimedia Learning Objects," *Proc. Fourth Australasian Computing Education Conf.*, ACM Press, New York, 2000, pp. 87-94.



Abdulmotaleb El Saddik is an assistant professor in the Department of Electrical Engineering at the Darmstadt University of Technology, Germany. His research interests include interactive multimedia environments, collaborative systems, multimedia

communications, and telecommunications software. He received his MSc (Dipl.-Ing.) and a PhD (Dr.-Ing.) in electrical engineering from the Darmstadt University of Technology, Germany. He is a member of the ACM, The Gateway to Educational Materials (GEM), and the *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning* editorial board.



Stephan Fischer is the chief technology officer at Mobile Video Communications in Frankfurt, Germany. His research interests are in content processing of video and audio and in cooperative multimedia learning systems. He received a diploma in business administration and computer science and his PhD in computer science from the University of Mannheim, Germany.



Ralf Steinmetz is a professor in the Electrical Engineering and Information Technology and Computer Science departments at the Darmstadt University of Technology, Germany, and has a chair position in the process communications and multimedia networking area. He is also a director of the Information Transfer Office at Darmstadt University. Since 1996, he has been director of the Integrated Publications and Information Institute of the German National Research Center (GMD) in Darmstadt, Germany. His research interests are in networked multimedia systems, cooperative applications, and mobile and service gateways for multimedia data. He has an MSc (Dipl.-Ing.) and a PhD (Dr.-Ing.) from the Technical University of Darmstadt, Germany. He is the associate editor in chief of *IEEE MultiMedia* magazine. He is a member of the ACM, Association of Computer Science (Gesellschaft fuer Informatik, GI), and Association of Information Technology (Informationstechnische Gesellschaft, ITG) and a senior member of the IEEE.

Readers may contact El Saddik at Industrial Process and System Communications, Dept. of Electrical Eng. and Information Technology, Darmstadt University of Technology, Merckstr. 25, D-64283 Darmstadt, Germany, email Abdulmotaleb.El-Saddik@kom.tu-darmstadt.de

For further information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.