[EFSt00] *Abdulmotaleb El Saddik, Stephan Fischer, Ralf Steinmetz;* **Reusability and Adaptability of Interactive Resources in Web-based Educational Systems;** ACM Journal of Educational Resources in Computing (JERIC), December 2000.

Reusability and Adaptability of Interactive Resources in Web-based Educational Systems

Abdulmotaleb El Saddik[1], Stephan Fischer[2] and Ralf Steinmetz[1,2]

1
Industrial Process and System Communications
Dept. of Electrical Eng. & Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany

2
GMD IPSI
German National Research Center
for Information Technology
Dolivostr. 15 • D-64293 Darmstadt • Germany

{Abdulmotaleb.El-Saddik,Ralf.Steinmetz}@kom.tu-darmstadt.de
sfischer@darmstadt.gmd.de

Abstract

The production of interactive multimedia content is in most cases an expensive task in terms of time and cost. It must hence be the goal to optimize the production by exploiting the reusability of interactive multimedia elements. Reusability can be triggered by a combination of reusable multimedia components, together with the appropriate use of metadata to control the components as well as their combination.

In this article, we discuss reusability and adaptability aspects of interactive multimedia content in web-based learning systems. In contrast to existing approaches, we extend a component-based architecture to build up interactive multimedia visualization units by the use of metadata for reusability and customizability issues.

In the three-tier model, the lowest layer of the paradigm corresponds to the programmer (code reusability). The user interface of an educational visualization is located at the top-layer medium where the interaction of the end-user (student) takes place. The educator is located between the top and the bottom layer. This medium layer allows for the adaptation of interactive multimedia content according to user's needs, applying a pre-defined set of metadata. The teacher can both adjust the level of explanation and the level of interactivity of an animation, hence influencing the presentation and the results of the algorithms being illustrated (program reusability). After a theoretical overview we explain our architecture by an application example.

1 Introduction

The rapid advancement in computer, communication, and presentation technology produces new forms of media and communications that can be used to increase the quality of educational documents to visualize complex technical problems. To help students learn difficult concepts, interactive learning software needs specific capabilities for simulation, visualization, and real-time data collection, as well as tools for analyzing, modeling and annotating data. Such interactive, dynamic representations are the core content of educational learning modules. These representations have to be combined flexibly with many kinds of contexts: Diverse classroom presentations, tutorials, experimentation notebooks and standardized assessments. To achieve that goal, the standardization of so-called *learning objects* became an important issue in the past.

As stated in the specification of IEEE's Learning Objects Metadata (LOM) [Gro00], "a learning object is defined as any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning". Examples of learning objects include multimedia content, instructional content, instructional software and software tools, referenced during technology supported learning. In a wider sense, learning objects could even include learning objectives, persons, organizations, or events. A learning object is not necessarily a digital object; however, the remainder of this paper will focus on learning objects that are stored in a digital format.

IEEE's learning object (LO) model is characterized by the belief that independent chunks of educational content can be created that provide an educational experience for some pedagogical purpose. With regard to object-oriented programming (OOP) and component-oriented programming (COP), this approach assumes that these chunks are self-contained, though they may contain references to other objects, and they may be combined or sequenced to form longer (larger, complex, other) educational units. These chunks of educational content may be of any type, interactive (e.g. simulation) or passive (e.g. simple animation), and they may be of any format or media type.

Another requirement for learning objects is related to tagging and metadata. To be able to use such objects in an intelligent fashion, they must be labelled as to what they contain, what they communicate, and what requirements with regard to their use exist. A reliable and valid scheme for tagging learning objects is hence necessary.

The LO model provides a framework for the exchange of learning objects between systems. If LOs are represented in an independent way, conforming instructional systems can deliver and manage them. The learning object initiatives, such as IEEE's LOM or Educom's IMS are a subset of efforts to creating learning technology standards for such interoperable instructional systems. In the same time repositories for educational object components have been built [Eoe00], [Gam00] and [Gem00]. These efforts gain leverage from the rise of interactive web technology and its associated emphasis on standards-based interoperability. Although the component-based solutions developed to date are useful [RDK+99], [GSvD99], they are inadequate for those 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. We see the development of techniques for sharing semantics across components and applications to be a critical research direction for the field.

The approach described in this paper addresses the issue of developing and customizing dynamic multimedia objects using *dynamic metadata*. We refer to these objects as *smart learning objects*. With the term *customization* we denote changes and/or modifications to a learning object. These changes are necessary to match the learning goals of a user and to reuse dynamic multimedia content in a different context. However, current versions of learning metadata do not address specific issues of dynamic content, such as interactivity or reusability. The main goal of this article is hence to examine the current learning metadata standards, and to propose extensions in order to match the specific constraints of multimedia content. As a starting point we address metadata extensions for visualizations and animations. Other dynamic metadata, for example those for movies, interactive environments (for example Macromedia Director objects), or for VRML-applications can be derived from those extensions.

To describe visualizations and animations appropriately, we introduce a new set of metadata which is an extension of IEEE's Learning Objects Metadata. We show how such metadata can be processed by a metadata editor, which allows us to describe smart multimedia objects. Another tool to customize the resource according to the user's needs will also be explained in the remainder of the article.

The article is structured as follows: In Section 2 we list related work and define multimedia Learning Objects (LOs) as well as learning objects metadata. In Section 3, we discuss the granularity of interactive multimedia content and their characteristics, before we introduce dynamic metadata in Section 4. Applications of dynamic metadata is described in Section 5, and an example is described in more detail in Section 6. Section 7 concludes the paper and gives an outlook.

2 Context

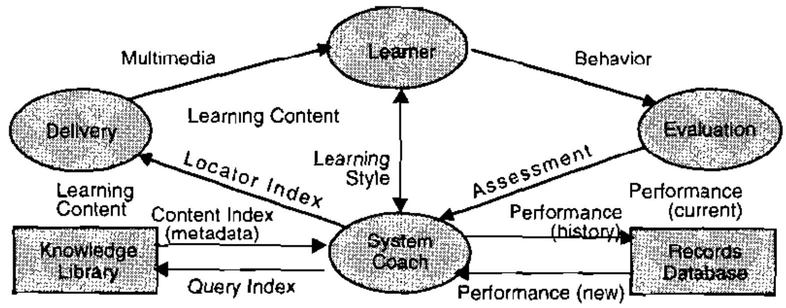
To explain our starting point and to communicate the motivation for our work, we first present an overview of the Multibook project currently developed at the Technical University of Darmstadt.

2.1 Multibook

The Multibook project [Mul00] currently being developed by the Technical University of Darmstadt and the Fern-Universität Hagen is a web-based adaptive hypermedia learning system for multimedia and communication technology. Multibook focuses on providing end-users with specific lessons tailored to a targeted group. These lessons are created using a knowledge base of multimedia elements, especially interactive animations. The creation of these lessons is done in an automatic fashion (course sequencing).

Multibook's knowledge base, which is necessary to implement the course sequencing [SSR+00], consists of two separated knowledge spaces. The *Concept Space* contains a networked model of learning topics [FS00] and uses approaches well known from knowledge management. The knowledge topics are interconnected via semantic relations. The media bricks stored in the *MediaBrick Space* of the system are atomic information units of various multimedia formats. These units are interconnected via rhetoric relations. Each media brick is described using IEEE's Learning Objects Metadata (LOM) scheme [Gro00]. In the following we refer to media bricks as *learning objects.* Although both information spaces are separated, each learning object can have a relation to one or more related topics. The separation of both spaces is the way in which Multibook generates adaptive lessons, because for each topic a set of media bricks (texts in different granularity, animations, video, etc.) is available. The selection of media bricks is then determined by the preferences of each user.

The general functionality of Multibook, in other words the way the sequencing of lessons, is based on the knowledge base stored in the Concept Space. This approach is similar to the standardization by IEEE as IEEE proposes the use of a knowledge library (knowledge base) which is responsible for the sequencing of a lesson, while the actual compilation of the lesson is performed by a delivery component (see Figure 1). It is essential to understand the setup of our knowledge base in order to understand the automatic creation of exercises.



The architecture of MultiBook is presented in Figure 2. It should be noticed that the architecture is very similar to the one proposed by the IEEE LOM group.

Considering the way an author writes a document the following order can be specified: (1) an author acquires background knowledge, (2) an author creates an outline for a document, (3) an author fills the outline with content. These steps are modeled by different spaces in Multibook. The Concept Space contains an ontology in terms of keywords which is necessary to create the outline of a lesson. After the sequencing of the outline (equally to the creation of a table of contents) the "real" content (text, images, audio, video, animation) is filled into the outline using elements of the second space, the MediaBrick Space. A general idea of Multibook is that it is necessary to employ different relations within the Concept Space and the MediaBrick Space to model the different goals which both spaces have.

When working with media bricks and with the necessary educational metadata, an important disadvantage becomes obvious: Due to the history of the development of metadata, static resources, such as images or text documents can be described properly. Unfortunately, an appropriate description of dynamic resources, for example animations, is feasible only to a limited extent. The reason is that dynamic multimedia objects can process input parameters, generate output parameters, and also work internally with data which cannot be described with traditional metadata schemes.

2.2 Interactive Multimedia Content and Static Metadata

With respect to adoptability, learning systems can be divided into two categories:

- Systems which deploy learning objects that are relatively simple but richly tagged using metadata. Each learning system operates on metadata with a significant degree of own knowledge.
- Systems which use learning objects that are smart in a sense that they can change their behavior. The system has to pass specific information to a LO, and each LO has to adhere to a particularly stipulated set of input/output parameters.

An example of the first category is the use of IEEE's LOM in the Multibook project in order to describe multimedia content. However, multimedia content being part of learning systems can be text, graphics, audio, video, animation, or simulation. Multimedia learning objects can be characterized as follows:

- *Multicodality*:
  Use of various symbol systems, for example images, pictographs, texts, etc.
- *Multimodality*:
  LOs can address different senses: The visual, the aural sense, or both at the same time.
- *Dynamics*:
  LOs make use of discrete (text, images) or continuous media (video, animation)
- *Interactivity*:
  LOs realize to some extent the interaction between learner and learning system.

Figure 3 illustrates the above mentioned characteristics and their relationships. An active map object belongs for example to the intersection of multicodality and interactivity. The common denominator of all these characteristics is what we refer to as a *Smart Learning Object*. Simulations which visualize complex procedures dynamically and interactively, belong to the group of smart learning objects. The use of animated graphics or simulations is much closer to real life than still graphics are. Complex procedures can be experienced, understood and learned by experimenting in the virtual environment being offered by simulations. The behavior of smart learning objects can be changed, as well as adapted according to parameters which are passed by the system. For the remainder of this paper we will denote interactive visualizations as *smart learning objects*.

One of the key problems in developing learning software systems in general and interactive instructional visualization units in particular is the integration of user requirements changing over time. Learning systems must be flexible in that they must be easy to adapt to new and changing user requirements.

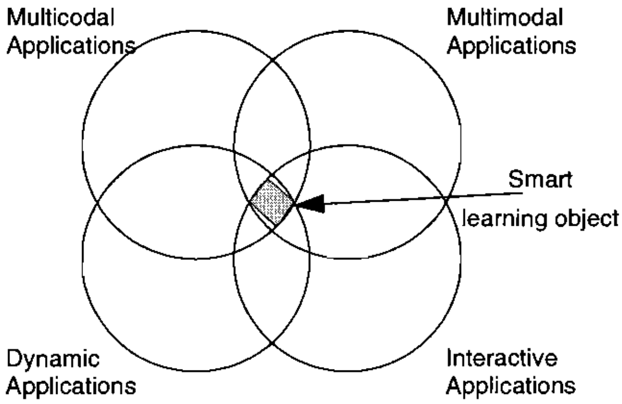


Figure 3: Characteristics of learning objects

An ideal scenario of software engineering is to build applications by putting together high-level components [RDK+99], [SSS+00], [KH97], and [GSvD99]. If any necessary components are not found, they have to be composed out of existing lower-level components. The functionality of such a system can then be changed or extended by substituting or plugging in new components. The components of the lowest-level have to be implemented in a certain programming language. Component software can be viewed as self-contained, pre-fabricated, pre-tested, black-box reusable software units. Example of component software is the JavaBeans technology.

3 Granularity of Smart Learning Objects

3.1 Coarse-grained SLOs

As a part of the Multibook project we developed several SLOs based on coarse-grained modules, which visualize specific algorithms. These modules have been implemented as Java applets that illustrate concepts and algorithms of multimedia communication technology. We use these components for the multimedia communication courses [http://www.kom.e-technik.tu-darmstadt.de/Teaching/Visualization/visualization.html] being taught at the Technical University of Darmstadt. They are built as units with a broad focus, illustrating many different sub-topics of the concept they were designed to visualize and teach for, allowing students to experiment with many different combinations of parameters.

The main problems with coarse-grained SLOs is that they illustrate only some of the intermediate computations that take place within an algorithm. This approach is appropriate for certain concepts, mainly for those courses where self-contained algorithms and data structures are taught. 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 [SN95] or MPEG [LPFL97] serves as a good example: Even though both compression schemes use the Discrete Cosine Transform (DCT) and the Huffman encoding, a reuse of a component of an animation of JPEG can in most cases not be used to visualize a step of the MPEG-compression process, if the illustration is coarse-grained. Coarse-grained animations are very useful in demonstrating the final concept, but are hard to use in teaching the individual ideas that are part of that concept.

3.2 Component-Based Development

Other problems of interactive multimedia animations are the video-like nature of animations, the wiring of animation components, the reusability of such components, and their structure. To address these problems and the issues mentioned above we developed a fine-grained approach, developing applets that are composed of small, atomic units. We developed a component-based framework [SSS+00] in order to generate complex animations based on simple modules, which visualize the different steps of an algorithm (JavaBeans technology). The exact granularity of the developed modules is strongly correlated with the domain being addressed and varies widely between concepts. Our goal with the modular model was to strive for the smallest possible scope for each concept.

A great advantage of the modular approach is that several fine-grained applets can serve as stand-alone applets illustrating individual ideas. They can also be reused and combined with others in order to visualize a more complex topic.

To be able to integrate these animations into Multibook, and in order to optimize their utilization, we tagged all animations using IEEE's Learning Objects Metadata.

As mentioned above, LOM can be used to search, navigate, and adapt the content of Multibook as long as static learning objects are used. However, the particular potential of interactive visualizations, in other words their flexibility and adaptability, can only be exploited to a limited extent. For example, some interactive visualizations can be used to illustrate different scenarios or different parts of an algorithm, depending on the input parameters. The same learning object can be re-used in a different learning context, according to the way it is configured by parameters. Parametrization of interactive visualizations can be done off-line or on-line. In order to achieve an on-line customization, we propose the use of dynamic metadata as an extension of the static IEEE Learning Objects Metadata.

4. Metadata for Dynamic Learning Objects

4.1 Multimedia Learning Objects Metadata

The starting point for our research is the existing technologies, standards, and on-going initiatives with regard to multimedia educational metadata. The Dublin Core [DC00] Metadata Element Set, Educom's Instructional Management System (IMS) [IMS00], the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) [IMS00], and IEEE's Learning Object Metadata Working Group 12 [Gro00] are the most important initiatives dealing with metadata for computerized learning purposes. These initiatives are closely related to the Resource Description Framework (RDF) [RDF00], the Warwick Framework [War96], and to other activities of the World Wide Web Consortium.

All of the methods used to specify metadata make use of metadata in the traditional sense of describing static data:

- to summarize the meaning of the data (i.e. what is the data about).
- to allow users to search for data.
- to allow users to determine if the data is what they want.
- to prevent some users (e.g. children) from accessing data.
- to retrieve and use a copy of the data (i.e., where do I go to get the data).
- to instruct how to interpret the data (e.g. format, encoding, encryption).

That is, the metadata descriptors are associated with the data sets in a fixed way. Their granularity is as defined by the original metadata author.

A great drawback is that the application of metadata is mainly limited to the above described fields where metadata are used in a static way with respect to the content. A first observation is that such metadata cannot describe smart dynamic LOs adequately. Metadata can also not influence the multimedia content itself, because metadata usually contain universal and widely applicable descriptions of objects. In our point of view, the usage of dynamic multimedia learning objects, such as animations, requires a new sort of metadata, which must be dynamic in order to facilitate the I/O behavior of a dynamic LO. In the following we will discuss the definition of smart learning objects before we define the necessary set of these metadata.

4.2 Dynamic Metadata

We define the term "dynamic metadata" as the description used to adapt the content of an object, and/or to change the behavior of a learning object.

As an example of dynamic metadata, we will in the following examine the simulation of the CSMA/CD protocol (Ethernet). To be able to explain Ethernet properly, specific problems have to be addressed, for example the collision of packets on the bus, or the shortframe problem. The key idea behind dynamic metadata is that the same visualization can be used to explain different problems, if it is configured by parameters. In the following we will explain the data structures for dynamic metadata in detail, but to motivate the problem, we provide an example here. A part of the data structure could be a field "PROBLEM", addressing a specific parameter configuration of a visualization. Concerning the visualization of Ethernet, changing the value of the metadata field "PROBLEM" (being represented in the program as a property) from "Collision" to "Shortframe" may change the whole behavior of the algorithm to be visualized.

Extension of IEEE's LOM

As mentioned above we understand dynamic metadata as an extension of IEEE's Learning Objects Metadata. The scheme of dynamic metadata follows the generic format of <property, values, value type>. According to the LOM specification [Gro00], this scheme is illustrated in Figure 4.

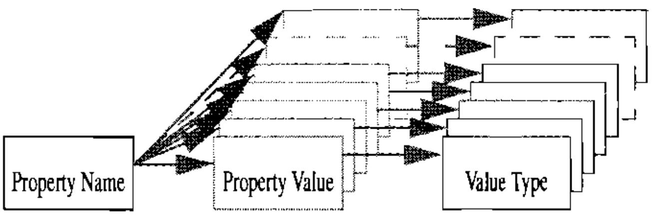


Figure 4: Generic scheme of dynamic metadata

In the following we analyze the requirements of the proposed set of metadata for dynamic content in detail.

- *Language*:
  LOM contains a field to store information about the language which is used within a LO. However, smart LOs enable the user to change that language. An example is Java's internationalization framework where a set of language alternatives can be used. Although it would be possible to change the underlying LOM base category, we propose to use a new field within the new category "Dynamic Metadata" containing a list of possible languages. The original LOM field could then be used to store the initial state of a smart multimedia LO.
- *DifficultyLevel*:
  Within the category "Educational", LOM contains the field "DifficultyLevel" that describes

the difficulty of a LO on a scale from "very low" to "very high". With regard to hierarchical modularized animation chains, such a choice is inappropriate. An example is an animation visualizing the steps of JPEG for a beginner. The level of difficulty would be "very low". A more advanced user could switch to animations of the single steps of the algorithm, an expert user could even change the components of the DCT formula. To be able to describe these difficulty changes, we introduce a new dynamic field "DifficultyLevel" which indicates the degree of difficulty the resource should start with. The values should (like in the LOM base model) range from "very low" to "very high". Modularized hierarchical animations have for example been described in [SSS+00].

- *InteractivityLevel*:
  The same argumentation with regard to the field "DifficultyLevel" is true for the degree of interactivity of a resource. While a beginner might use a visualization of a problem in a movie-like style, an expert might want to change parameters and thus use a highly interactive application. We propose a new field "InteractivityLevel", storing the degree of interactivity on a scale from "very low", "low", "normal", "high", to "very high".
- *Bidirectional*:
  Some animations or visualizations offer the possibility to step forward or back within a smart LO. We propose to use a field "Bidirectional" indicating whether a step-back operation is possible or not.
- *Dimension*:
  For some animations it is necessary to specify the dimension of the container in which the visualization will take place.
- *Topic*:
  Many smart LOs like animations or visualizations explain an algorithm with multimedia elements. We propose to use a field "algorithm" to store the name of an algorithm. Another possibility would be to extend the meaning of the field "name" of the base LOM scheme. The disadvantage of the latter approach would be that a clear distinction between the parts of an algorithm would be impossible. An example is a JPEG animation where the field "name" contains the string "JPEG", while the dynamic metadata field "algorithm" might contain the string "entropy encoding".
- *Scenario*:
  A scenario is a specific form of an animation which is defined by a teacher and intended to explain a subset of the knowledge a smart LO could transfer. Similar to the field "algorithm", a smart LO can be used to visualize various scenarios. An example is a smart LO explaining Ethernet. Possible scenarios are for example "shortframe" or "collision". The new field "scenario" has a general meaning as it can be identified in many smart LOs; it is somehow an alternative to the field "algorithm". To be able to describe a scenario adequately, we define two more new fields: "mode" and "name". In the mode field we offer a selection of the values "problem", solution", "interactive", and "guidedTour". These values can be used by a teacher to define the degree of interactivity he wants to assign to a resource. The name field stores the name of the respective scenario. The field "scenario" can then store choices of the things a smart LO can explain. The items of the lists can have a different degree of interactivity.
- *InputData*:
  A very important new field with regard to dynamic metadata is the field "InputData". Regular static resources don't need any input data. Smart Learning Objects can be parameterized by input data. The same smart LO can then be used to animate different topics. An example is given in Section 5. Input data can for example be stored in a serialized way in a file. The field "InputData" then contains the file name of the input data.

- *OutputData*:
  Like input data, a smart Learning Object can communicate with the outside world using output data which can be stored in a serialized way in a file. The field "OutputData" of the category "dynamic metadata" would then contain the name of the file.
- *Explanation*:
  Many smart multimedia learning objects come with some sort of explanation, for example a text motivating a problem, or an audio introduction explaining the screen setup, or the processing which is visualized in the smart LO.We propose to use a new dynamic category "explanation" with the fields "type" and "media". The type of an explanation can for example be "hints", "errors", "logs", or "information". The type "hints" can for example activate a hint narration of the topic to be visualized. The type "errors" could activate an error rendering of the topic to be visualized. In some cases it can be very useful to inform the user about errors which result from an incorrect use of parts of a smart LO. The type "logs" can activate a narration of the logs of a smart LO visualizing a topic which can be used to discover the history of the use of a smart LO. The type "information" can activate an explanation of the general steps of an algorithm to be visualized by a smart LO. Many animations which can be found nowadays don't use an explanation of the animation itself which makes it sometimes hard to use the animation. The field "media" contains information about the storage format of the available explanation. Possible values are "Text", "Audio", or "Video". As an example, a combination of "Information" and "Audio" stored in the fields of the category "Explanation" would explain the functionality of the animation using a prerecorded audio file.

The general structure of the dynamic metadata category extending the base LOM scheme is shown in Table 1.

No.	Property	Description	Example
1	*Code Information*	The information concerning the code of the smart LO	
1.1	codeName	The name of the start code of the sLO	a.class, a.flash, etc.
1.2	codeLocation	Denotes where the sLO s located	URI
1.3	codePackage	The name of the package or zip of the sLO	a.jar, a.zip, etc.
2	*Presentation Information*	The information concerning how the smart LO is to be presented	
2.1	Language	The language, the smart Learning Object (LO) should start with.	en, de, fr, etc.
2.2	DifficultyLevel	The degree of difficulty the resource should start with.	very low, low, normal, high, very high
2.3	InteractivityLevel	The degree of interactivity of the resource.	very low, low, normal, high, very high

Table 1: Proposed fields of metadata for dynamic content

2.4	Dimension	The 3 dimensions Information of the visualization unit.	x,y,z
2.5	Bidirectional	Indicates whether the explanation, visualization can be done in the back direction or not.	yes, no
3	*Topic Information*		
3.1	Topic	The name of the topic to be shown by the resource.	Fifo, Earliest Dead-line First,...
3.2	Scenario	The name of the scenario to be visual-ized by the resource.	-
3.2.1	Mode	Intention of the teacher with regard to the smart LO.	problem, solution, normal, guidedTour
3.2.2	Name	Stores the name of a scenario.	collision, shortframe
3.3	InputData	The name of the input file needed by the resource to start properly.	parameters.txt
3.4	OutputData	The name of the output file the resource should generate.	parameters.txt
4	*Explanation Information*	Indicates which kind of explanation is required for a smart learning object.	-
4.1	Type	A list of possible explanation types.	Information, warn-ing, error, log, hint
4.2	Media	A list of possible media types to be used for the explanation	Text, audio, video

Table 1: Proposed fields of metadata for dynamic content

It should again be noted that the LOM base scheme already introduced some fields which are similar to the ones described above. An example is the field "language". These fields are however not well suited to describe the special abilities of smart multimedia Learning Objects.

5 Application of Dynamic Metadata

The development of visualization instructional units using a component-based (fine-grained) approach in combination with metadata has both benefits and drawbacks. The benefits are flexibility, reusability, and effective use in several learning contexts according to user's needs. The drawbacks are the needs to provide supporting descriptive materials and essential modifications to software engineering practice.

5.1 Reusability

The programmer identifies animation objects that visualize the steps of an algorithm. He defines the smallest modular entities of the algorithm and develops them as black-box software components, which may be re-used within the development of other algorithms. The programmer specifies animation actions at particular locations in the algorithm chain, according to a predefined set of metadata. The programmer's view has been described in detail in [SSS+00]. In our project we use the JavaBeans technology to create a publicly available component library that can be used by all team members as well as by others to develop teaching animations. Beyond single components, entire fine-grained applets are reusable for different courses.

5.2 Adaptability, Flexibility, and Customizability

It is essential that a visualization is flexible enough to confront and address changing user requirements and knowledge. It should also be versatile, and usable in a variety of contexts. While often, a visualization will not fit the needs of a particular user "off the rack", it can be tailored to do so when certain "alterations" are done. This leads to the fact that it is easy to integrate animations in the right context. According to the needs of the end-user, the educator may convert an algorithm developed by the programmer to a series of animation sequences by mapping algorithm variables, specifying animation actions, and associating execution points in the algorithmic chain to perform the desired animation. He uses dynamic metadata, which have been pre-defined by the programmer.

5.3 Interactivity and Unified Graphical User Interface

Most of the animations being used to visualize complex algorithms and techniques can be influenced by the end-user. For the learner, a visualization window is divided into three areas: an animation pane, an explanation area, and a parameter pane. The animation pane displays the resulting animation envisioned by the educator. The explanation area displays some hints and information concerning the visualized algorithm. The parameter pane is divided into two parts: Interactive utilities and a VCR, allowing the learner to control the progress of the animation. The interactive utilities pane depends on the algorithm and on the topic, defined by the programmer and customized by the educator. The animation may request intermediate input from the learner, allowing him to control the path of the algorithm.

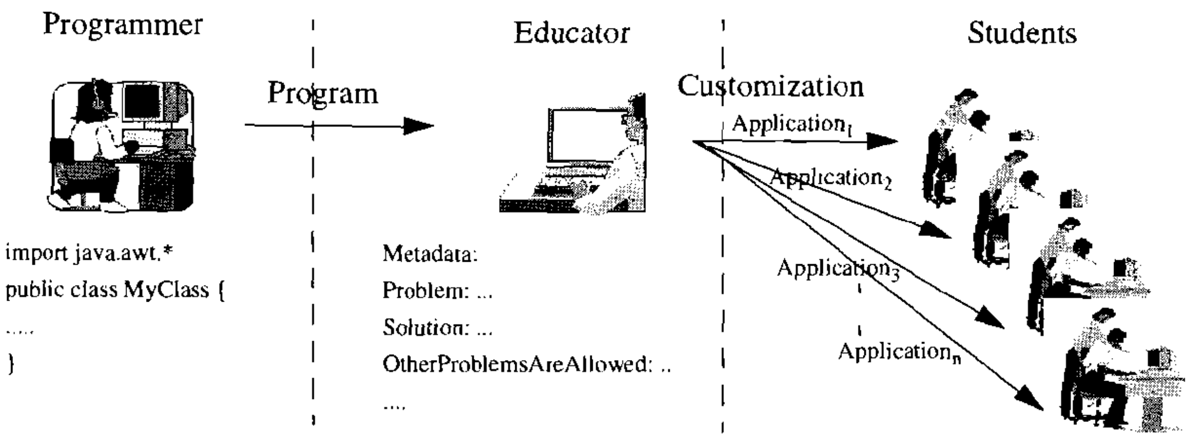


Figure 5: Reusability, flexibility, and interactivity

Figure 5 illustrate the process of developing, tagging, customizing and using interactive smart learning objects in our framework. The programmer is in charge of writing the appropriate code of the algorithm to be animated. The educator becomes the designer of the visualization to be shown. He can customize the learning object in order to visualize a desired behavior, which is appropriate for the course to be taught. The student becomes the end-user of the customized animated algorithm.

6 Application Example

Figure 6 shows the overall architecture of our smart Learning Object tagging and customizing architecture. Learning resources are tagged using a special metadata editor called xLOM editor (see Section 6.1). For the storage of static and dynamic metadata we use a relational database (see Figure 2). To access the data stored in the SQL database we developed a three tier architecture using JDBC.

We also implemented a tool to customize interactive visualizations with the use of dynamic metadata. We call this tool, which is described in more detail in Section 6.2, "content customizer". We use the content customizer to customize a smart Learning Object in different ways for the specific purposes of a lesson. We are then able to use visualizations several times in a learning unit, according to the context of the unit, which is described in detail in Section 6.3. In Figure 6, a smart Learning Object is reused within different scenarios with different metadata sets to show different scenarios of the same topic.

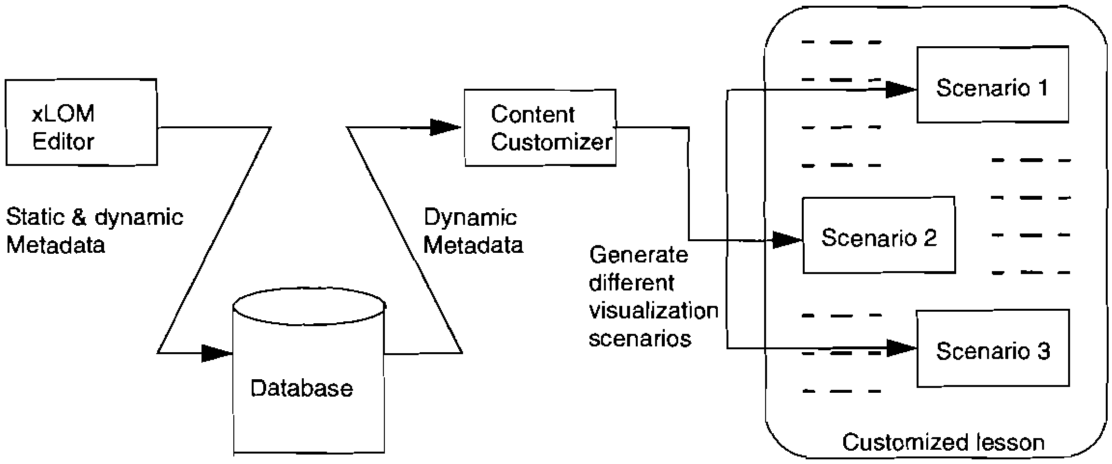


Figure 6: Smart Learning Objects tagging and customization process

6.1 xLOM Editor

In the following we describe the tool which we use to create both static and dynamic metadata. The tool can also be used to publish metadata records for various resources, e.g. documents, images, audio clips, videos, animations, virtual reality worlds, or multimedia exercises.

A metadata record consists of a set of elements, describing a multimedia resource. Examples of these elements are *date of creation or publication*, *type*, *author*, *format*, or *title* of a resource. To access and discover multimedia information resources in a comfortable way, we developed a tool, the LOM editor, based on the IEEE-LOM scheme version 4.0. The LOM editor can be used to create and store LOM records in a relational database, and can also be used to query the database and to navigate on a resulting metadata set. When working with the editor, it turned out quickly that smart LOs can only be described to a limited extent using the base LOM scheme. We extended the LOM editor to a new editor called *xLOM* (Extended LOM) *editor* by adding an extra category for dynamic metadata which has been described in Section 4.2

When tagging the source material with the xLOM editor, an interesting experience turned out: Most elements of a lesson to be described apply the same basic metadata information, such as the *name of the author*, the *rights of the lesson*, or the *targeted user group*. It would hence be very useful to use a set of templates to tag the material. Templates can avoid the necessity to fill a lot of fields again and again, for example the owner fields, the necessary browser requirements, and many more. In our current implementation, templates are used to store information, which is then only typed in once and can be applied multiple times.

To be able to exchange metadata with other applications, we included an XML-based import/export functionality as part of the xLOM editor. This work is based on the LOM object model [IMS00] provided by IMS.

We used the xLOM editor to tag various multimedia elements, for example the Java applets that were developed as part of the Multibook project. An example of the new functionality introduced by our xLOM scheme is the animation explaining IEEE-802.3 Ethernet [Eth00], which will be explained in Section 6.3

6.2 Content Customizer

As discussed before, dynamic metadata contain properties of smart multimedia learning objects, and a set of suitable values for each property. By customizing smart objects, exactly one of these values will be assigned to its related property. An example is the dynamic metadata field "scenario" where the content customizer allows to select a mode (for example guided tour), together with a name of a scenario which has been stored before. To be able to offer this functionality, we implemented a content customizer tool.

Once a smart learning objects has been retrieved, the content customizer loads its dynamic metadata from the database. An interactive visualization can then be parametrized by selecting the necessary values for each parameter. The result can be stored in XML or HTML syntax.

6.3 Example

As an application example we developed a lesson explaining the protocol CSMA/CD (Ethernet). The main goal of the lesson is to demonstrate the possibilities that the parametrization of an animation offers. A German version of the lesson can be found on the web using the URL http://www.kom.e-technik.tu-darmstadt.de/projects/iteach/itbeankit/applets/paradelektion/.

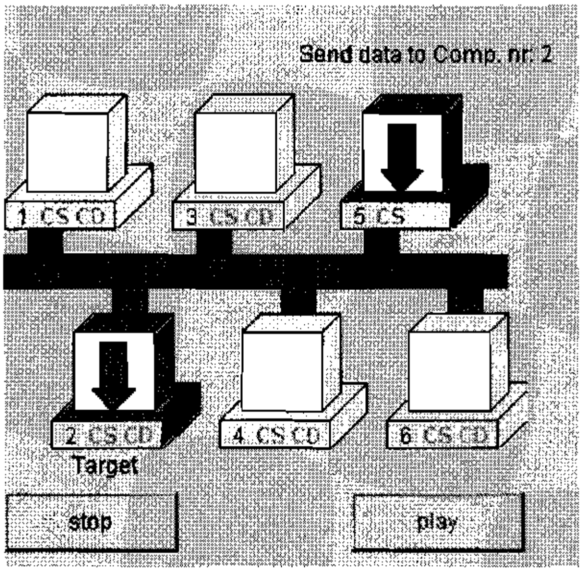


Figure 7: Example of parametrization of smart LO

In the example the functionality of Ethernet is explained first. After that the student has to answer the question which problems have to be faced in a bus-topology. We provide a set of different answers and use the same animation to explain why the answers are correct or wrong.

The difference between the answers is explained by different parametrizations of the same animation. These parametrizations are stored as dynamic metadata for smart multimedia learning objects as explained above. An example is shown in Figure 8 where the possible answer to the question "Why is the protocol complex" is "The protocol is complex because messages cannot be sent to a specific computer, they can only be sent to several computers at once."

Another scenario of the same animation can also be used to provide an answer to the question "How can the collision problem be solved in Ethernet" if dynamic metadata are used. A possible answer to the question would be "Collisions cannot be avoided. If a collision is detected the transmission has to be repeated". A parametrized version of the same animation used in Figure 7 is illustrated in Figure 8. Note that due to the use of dynamic metadata the same animation is reused in a different context.

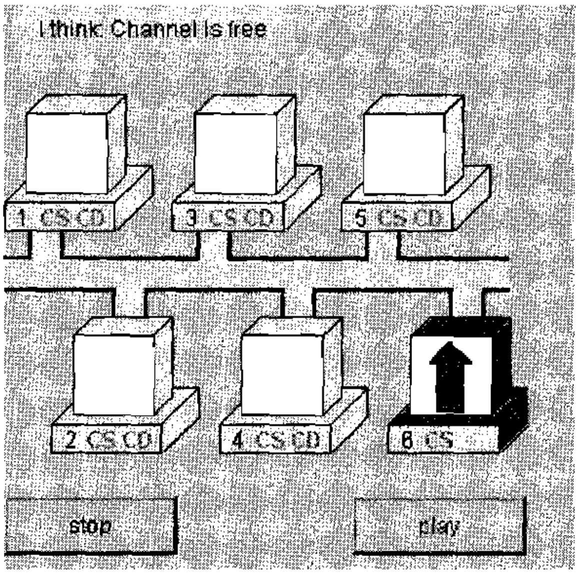


Figure 8: Second example of parametrization of smart LO

A screenshot of the lesson is shown in Figure 9.

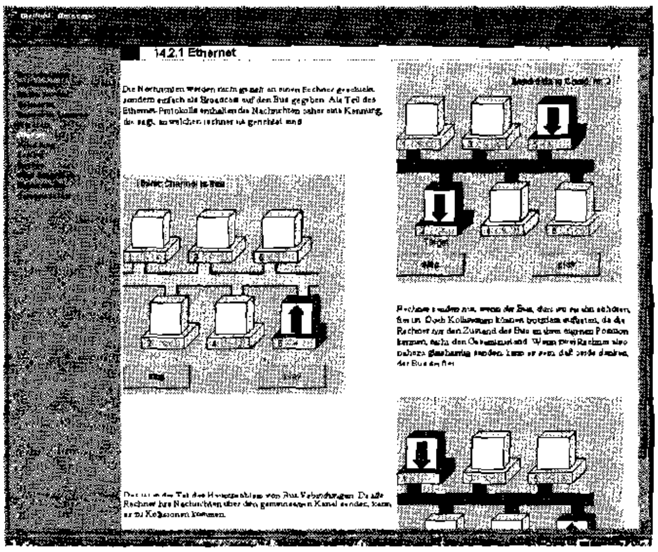


Figure 9: Screenshot of example lesson

7 Conclusion and Future Work

In this paper, we discussed the reusability aspects of multimedia content in web-based learning systems. We highlighted the necessity to develop component-based interactive multimedia visualization units, and we suggest the use of metadata for reusability issues. The main contribution of the article is an extension of IEEE's Learning Objects Metadata, to be able to describe dynamic multimedia learning objects to which we refer to as *smart multimedia learning objects*. Traditional metadata to describe learning objects are well suited to describe static elements (for example texts, or images), but do not take into account the dynamic nature of multimedia elements (especially animations and multimedia presentations). We hence compared various learning metadata standards and derived an extension which is suited to solve the problem.

The metadata-based framework presented in our article also addresses the customization of smart learning objects by metadata. Having explained the necessary category of dynamic metadata, we described our implementation of the tools which can be used for tagging, storing, and customizing smart Learning Objects. As a prototype, we implemented visualization artifacts dealing with the network protocol "CSMA/CD (Ethernet)". The implementation is also available on the web: http://www.kom.e-technik.tu-darmstadt.de/projects/iteach/itbeankit/applets/paradelektion/. We are currently using our framework to develop other examples of teaching animations for our multimedia and communications courses, for example animations to explain multimedia scheduling algorithms.

The research described here differs from other related work in that the set of dynamic metadata items that can be defined for a *Smart Learning Objects* is open ended and not fully predefined. The predefinition of all attributes would hinder the ability to support multiple applications. Instead, users are allowed to create whatever metadata records are necessary to support the object customization process, together with a necessary base set of pre-defined parameters which describe dynamic resources.

We understand our article as a starting point towards the development of metadata which are suitable to describe any type of instructional content, extending the traditional understanding of static metadata. We are well aware of the fact that such a standardization is a complex process and would like to initiate a discussion in that area.

8. Acknowledgments

The authors would like to thank the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) and the Volkswagen Stiftung who partially funded the research project.

9 References

[DC00] Provisional report of the dublin core subelement working group. http://purl.oclc.org/metadata/dublin_core/wsubelementdrafts.html, 2000.

[Eoe00] Educational object economy. http://www.eoe.org, 2000.

[Eth00] ANSI/IEEE standard IEEE-802.3: Ethernet, 2000.

[FS00] S. Fischer and R. Steinmetz. Automatic creation of exercises in adaptive hypermedia learning systems. In Proceedings of *10th ACM conference on Hypertext*, 2000.

[Gam00] Gamelan educational applets page. http://developer.earthweb.com/directories/pages/dir.java.educational.html, 2000.

[Gem00] The gateway to educational materials. http://www.geminfo.org/, 2000.

[Gro00] IEEE Learning Technology Standards Committee (LTSC) IEEE P1484.12 Learning Objects Metadata Working Group. http://ltsc.ieee.org/wg12/, 2000.

[GSvD99] Daniel L. Gould, Rosemary M. Simpson, and Andries van Dam. Granularity in the design of interactive illustrations. In Proceedings of *ACM SIGCSE*, New Orleans, LA, 1999.

[IMS00] Educom's instructional management systems project (IMS). Website http://www.imsproject.org, 2000.

[KH97] R. Klein and F. Hanisch. Using a modular construction kit for the realization of an interactive computer graphics course. In Proceedings of *EdMedia'97*, 1997.

[LPFL97] Joan L.Mitchell, William B. Pennebaker, Chad E. Fogg, and Didier J. LeGall. *MPEG video compression standard.* Chapman &Hall, 1997.

[Mul00] The multibook project homepage. http://www.multibook.de, 2000.

[RDF00] Resource description framework (RDF). http://www.w3.org/RDF/, 2000.

[RDK+99] J. Roschelle, C. DiGiano, M. Koutlis, A. Repenning, J. Phillips, N. Jackiw, and D. Suthers. Developing educational software components. *IEEE Computer*, 32(9):50–58, September 1999.

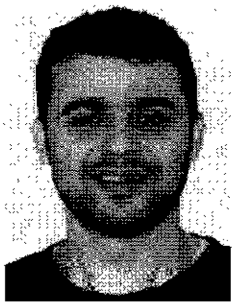
[SN95] Ralf Steinmetz and Klara Nahrstedt. *Multimedia computing, communications, and applications.* Prentice Hall, 1995.

[SSR+00] Cornelia Seeberg, Achim Steinacker, Klaus Reichenberger, Abdulmotaleb El Saddik, Stephan Fischer, and Ralf Steinmetz. From the user's needs to adaptive documents. In Proceedings of *IDPT*, Dallas, June 2000.

[SSS+00] Abdulmotaleb El Saddik, Cornelia Seeberg, Achim Steinacker, Klaus Reichenberger, Stephan Fischer, and Ralf Steinmetz. A component-based construction kit for algorithmic visualizations. In Proceedings of *IDPT*, Dallas, June 2000.

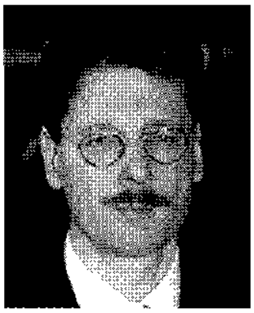
[War96] Carl Lagoze. The warwick framework, a container architecture for diverse sets of metadata. d-lib magazine, july/august 1996. http://www.dlib.org/dlib/july96/lagoze/07lagoze.html, 1996.

Author Biographies



Abdulmotaleb El Saddik is pursuing a Ph.D. in electrical engineering in the Department of Electrical Engineering and Information Technology at Darmstadt University of Technology, Germany. He is currently working on the Multibook project, founded by the German Federal Ministry for Education and Research (BMBF). His research interests include interactive multimedia visualizations, collaborative learning environments, multimedta communications, and telecommunications software. A. El Saddik received his M.Sc. (Dipl.-Ing.) degree in electrical engineering from Darmstadt University of Technology in 1995. He is a member of the Association for Computing Machinery (ACM), The Gateway to Educational Materials (GEM) and the Interactive Multimedia Electronic Journal of Computer-Enhanced Learning (IMEJ). He is also co-author of the Open Java book, published 1999 by Springer.

Stephan Fischer received a diploma in business administration and computer science from the University of Mannheim, Germany in 1994 and the Ph.D degree in computer science from the University of Mannheim, Germany in 1997. From 1997-1999, he was with the Department of Electrical Engineering of the Technical University of Darmstadt, heading the multimedia learning and content processing groups. From 1999-2000, he was with GMD-IPSI in Darmstadt. Since 2000, he is Chief Technology Officer at Mobile Video Communications (MVC) in Frankfurt. His research interests are mainly in content processing of video and audio, and in cooperative multimedia learning systems.



Ralf Steinmetz is professor at the "Electrical Engineering and Information Technology" as well as "Computer Science" departments of the Darmstadt University of Technology, Germany, since 1996. There he is in charge of a new chair position in the area of process communications and multimedia networking sponsored mainly by the Volkswagen-Stiftung. He is also one of the directors of the Information Transfer Office at the university. In late 1996 he was appointed director of the Integrated Publications and Informations Institute of the German National Research Center (GMD) in Darmstadt, Germany. Here he is in charge of co-operative work, workspaces of the future, interactive learning, media processing and mobile networking. His research interests are networked multimedia systems, co-operative applications, as well as mobile and service gateways for multimedia data. Dr. Ralf Steinmetz studied electrical engineering with the focus on communications at the University of Salford, England, and at the Technical University of Darmstadt, Germany, where he received the M.Sc. (Dipl.-Ing.) degree in 1982. Working as scientific assistant, he received the Ph.D. degree (Dr.-Ing.) in 1986 at this university. He mainly worked in the area of Petri-Nets and concurrent programming languages. Subsequently he joined the "Advanced Development Department" of "Philips Kommunikations Industrie" in Siegen-Eiserfeld, Germany, where he was involved in ISDN multimedia project workstations development activities. From 1988 until 1996 he worked at the IBM European Networking Center in Heidelberg, Germany. There he has been involved in various multimedia communication activities. He started his first activities on multimedia and networking and was in charge of a multimedia laboratory. In integrated multimedia projects that followed he acted as a key technical coordinator. He was the leader of the whole multimedia transport system development and subsequently was in charge of several application projects and their respective application support issues. He managed the multimedia department in Heidelberg which at that time became IBM's European Multimedia Center. He is editor and co-author of a multimedia course, which reflects the major issues of a first in-depth technical book on multimedia technology, 1993, (in German). He is editor of the magazine "Computer Communications" published by Butterworths-Heinemann and "Distributed Systems Engineering". He was in charge of a worldwide IEEE Multimedia Taskforce working group for magazine publication. There he is associate-editor in-chief of the IEEE Multimedia Magazine. He has served as chair, vice-chair and member of various program and steering committees of multimedia workshops and conferences. He is a member of ACM, GI, ITG, as well as senior member of IEEE.