[ES01] *Abdulmotaleb EL Saddik, Ralf Steinmetz;* **Keep It Simple and Smart; ACS/IEEE International Conference on Computer Systems and Applications ,Beirut, June 2001.**

Abdulmotaleb EL Saddik and Ralf Steinmetz. Keep It Simple and Smart. In Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, June 2001. pp:538-544

Keep it Small and Smart

Abdulmotaleb El Saddik[1] 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

http://www.kom.e-technik.tu-darmstadt.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.

1. Introduction

The rapid advaneement in computer, communication, and presentation technology produces new forms of media and communications that can be used to inerease the quality of educational doeuments to visualize complex technical problems. To help students learn diffieult concepts, interaetive learning software needs speeific capabilities for simulation, visualization, and real-time data colleetion, as well as tools for analyzing, modeling and annotating data. Such interactive, dynamic representations are the eore eontent 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 aehieve that goal, the standardization of so-ealled *learning objects* beeame an important issue in the past.

As stated in the specification of IEEE's Learning Objeets Metadata (LOM) [11], "a learning object is defined as any entity, digital or non-digital, whieh can be used, re-used or referenced during technology-supported learning" Examplcs of learning objects include multimedia content, instructional content, instructional software and software tools, referenced during technology supported learning. In a wider sense, learning objects eould even include learning objectives, persons, organizations, or events. A learning object is not necessarily a digital objeet; however, the remainder of this paper will focus on learning objects that are stored in a digital format.

IEEE's learning objeet (LO) model is characterized by the belief that independent ehunks of edueational content ean 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-eontained, though they may contain references to other objects, and they may be eombined or sequenced to form longer (larger, complex, other) educational units. These chunks of educational eontent 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 objeets 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 objeets is henee necessary.

The LO model provides a framework for the exchange of learning objects between systems. If LOs are represented in an independent way, eonforming instructional systems can deliver and manage them. The learning objeet initiatives, such as IEEE's LOM or Edueom's IMS are a subset of efforts to creating learning teehnology standards for sueh interoperable instructional systems. In the same time repositories for educational object components have been built [2], [9] and [10]. These efforts gain leverage from the rise of interactive web technology and its assoeiated emphasis on standards-based interoperability. Although the component-based solutions developed to date are useful

[18], [12], they are inadequate for those building eomponent-based interaetive learning environments in whieh the eomponents must respond to the meaning of the content as well as its form and presentation. We see the development of tcchniques for sharing semantics aeross components and applieations to be a eritieal researeh dircction for the field.

The approach dcscribed in this paper addresses the issue of dcveloping and customizing dynamic multimedia objccts using *dynamic metadata*. We refer to thcse objects as *smart learning objects*. With thc term *customization* we denote ehanges and/or modifications to a leaming object. These changes arc ncccssary to match the leaming goals of a user and to reuse dynamie multimedia contcnt in a differcnt context. Howcver, current versions of leaming metadata do not address specific issues of dynamic content, such as interactivity or reusability. The main goal of this article is hence to cxamine the current leaming metadata standards, and to propose extensions in order to mateh the specific constraints of multimedia eontcnt. As a starting point we address metadata extensions for visualizations and animations. Other dynamic metadata, for example those for movies, interaetive environments (for example Macromedia Director objects), or for VRML-applieations can bc dcrived from those extensions.

To deseribe visualizations and animations appropriately, we introduce a new set of metadata whieh is an cxtension of IEEE's Learning Objeets Metadata. Wc show how such metadata can be proeessed by a metadata editor, which allows us to deseribe smart multimedia objects. Anothcr tool to customizc the resource according to the user's needs will also bc explained in the remainder of the artiele.

The article is structured as follows: In Section 2 we list related work and explain the context of our work. In Scction 3, we discuss the granularity of interaetive multimedia content and their characteristics, before wc describe some applications of dynamic metadata. An example is than deseribed in more detail in Section 5. Section 6 concludes the paper and gives an outlook.

2. Context

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

2.1 Multibook

The Multibook projeet [16] currcntly being developed by the Teehnieal University of Darmstadt and the Fern-Universität Hagen is a web-based adaptive hypermedia learning system for multimedia and eommunieation teehnology. Multibook focuscs on providing end-users with specific lessons tailored to a targeted group. Thcse lessons are ereated using a knowledge base of multimedia elements, espeeially interactivc animations. The creation of these lessons is done in an automatie fashion (eourse scquencing).

Multibook's knowledgc base, which is neeessary to implement the course sequencing [20], eonsists of two separated knowledge spaces. The *Concept Space* eontains a networked model of learning topies [8] and uses approaches well known from knowledge management. The knowledge topics are intereonneeted via semantie relations. The media bricks stored in the *MediaBrick Space* of the system are atomie information units of various multimedia formats. These units are interconnected via rhetorie relations. Each media brick is described using IEEE's Leaming Objects Mcradara (LOM) scheme [11]. In the following we refer to media bricks as *learning objects*. Although both information spaces are separated, each leaming object can have a relation to one or more related topics. The separation of both spaees is the way in whieh Multibook generates adaptive lessons, because for eaeh topic a set of media bricks (texts in different granularity, animations, video, etc.) is availablc. The selection of media bricks is thcn determined by the preferences of each user.

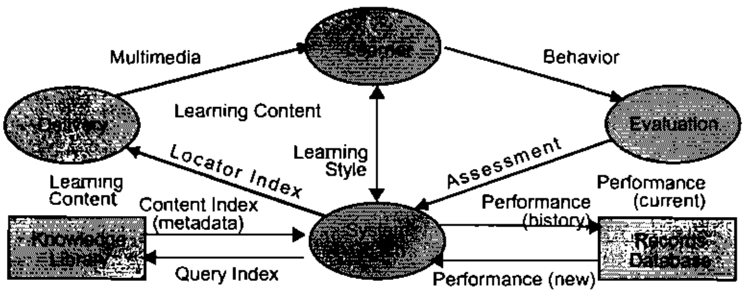


Figure 1: IEEE-LTSC-architecture

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 aetual 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-LTSC (Learning Technology Standards Committee) Learning Object Metadata 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.

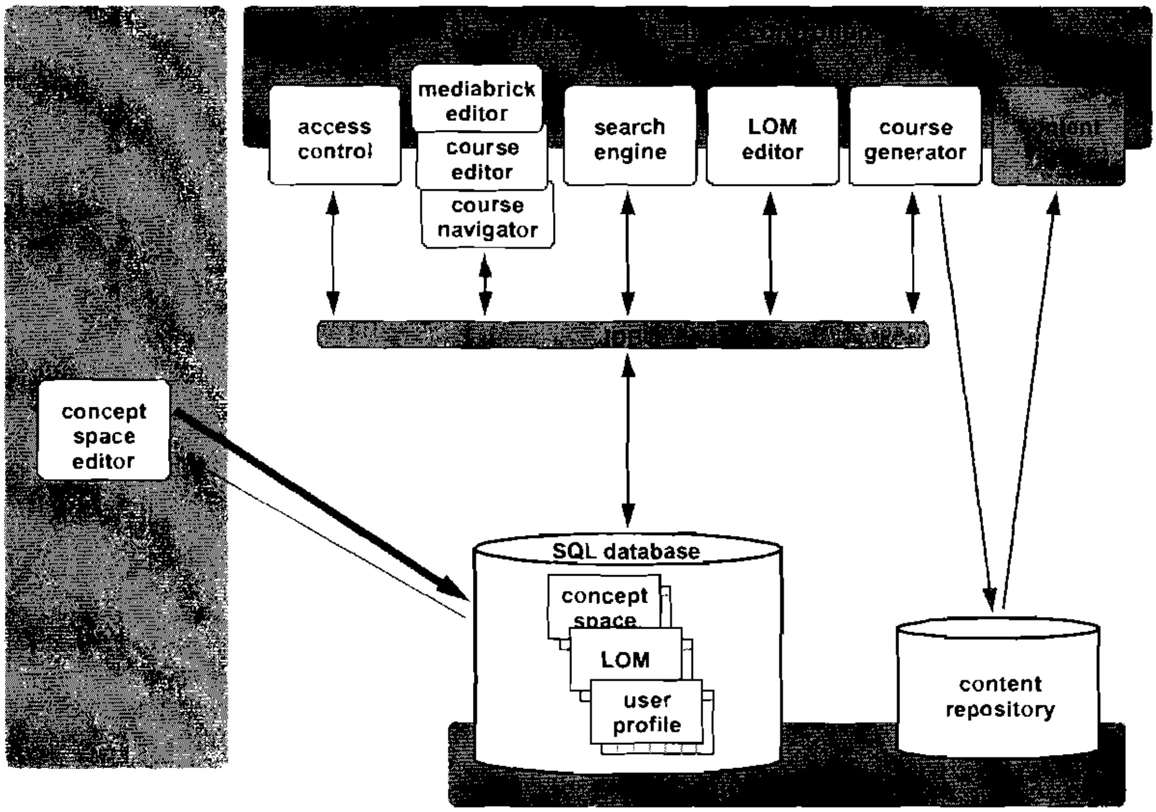


Figure 2: MultiBook architecture

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.

3. Granularity of Smart Learning Objects

3.1 Coarse-grained SLOs

As a part of the Multibook project we developed several (Smart Learning Objects) 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 [19] or MPEG [15] 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 [6] 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 proposed the use of dynamic metadata as an extension of the static IEEE Learning Objects Metadata [4].

3.3 Dynamic Metadata

We define the term "dynamic metadata" [4], [5] and [6] 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.

4. Application of Dynamic Metadata

The development of visualization instructional units using a component-based (fine-grained) approach in eombination with metadata has both benefits and drawbacks. The benefits are flexibility, reusability, and effeetive 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.

4.1 Reusability

The programmer identifies animation objeets that visualize the steps of an algorithm. He defines the smallest modular entities of the algorithm and develops them as black-box software eomponents, which may be re-used within the development of other algorithms. The programmer speeifies 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 [7]. In our project we use the JavaBeans technology to ereate 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.

4.2 Adaptability, Flexibility, and Customizability

It is essential that a visualization is flexible enough to confront and address ehanging 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 partieular user "off the rack", it can be tailored to do so when certain "alterations" are done. This leads to the faet that it is easy to integrate animations in the right eontext. 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 algorithmie chain to perform the desired animation. He uses dynamic metadata, which have been pre-defined by the programmer.

4.3 Interactivity and Unified Graphical User Interface

Most of the animations being used to visualize eomplex algorithms and techniques ean 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 eoneerning 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 edueator. The animation may request intermediate input from the learner, allowing him to control the path of the algorithm.

Figure 3 illustrate the proeess of developing, tagging,

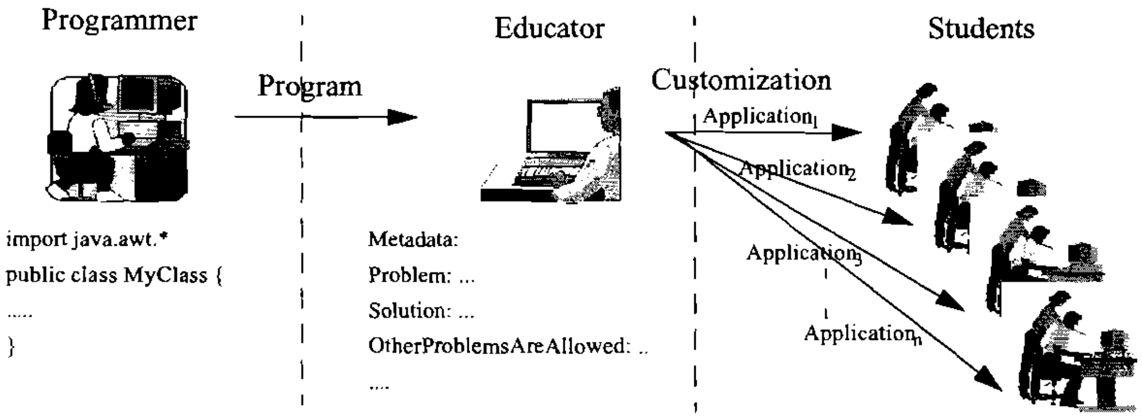
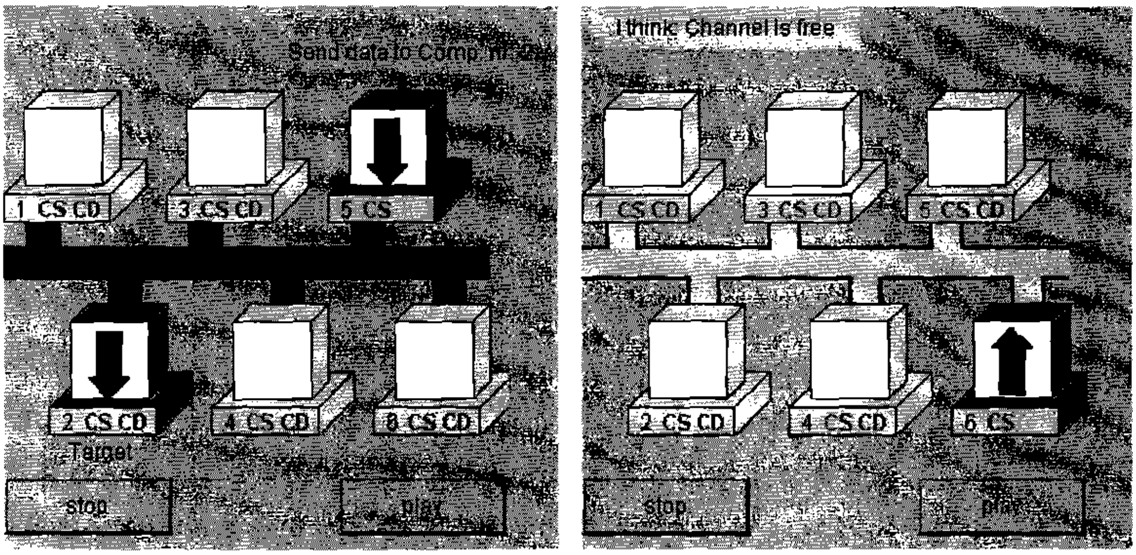


Figure 3: Reusability, flexibility, and interactivity

customizing and using interaetive smart learning objects in our framework. The programmer is in eharge 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 objeet in order to visualize a desired behavior, which is appropriate for the course to be taught. The student beeomes the end-user of the customized animated algorithm.

5. Application Example

The lesson is implemented in HTML 4.0 and contains visualizations developed according to the iTBeanKit framework enhanced with the dynamic metadata set described in [5] and [7]. The lesson consists of 15 pages. In the lesson the functionality of Ethernet [3] 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 parametrization of the same animation. These parametrizations are stored as dynamic metadata for interactive visualizations as explained above. An example is shown in Figure 4 (a) 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."



(a) (b)

Figure 4: Examples of parametrization of dynamic media brick

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". If a collision occurs, the transmitting station recognizes the interference on the network and transmits a bit sequence called jam. The jam helps to ensure 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.

A parametrized version of the same animation used in Figure 4 (a) is illustrated in the same figure to the right (b). Note that due to the use of dynamic metadata the same animation is reused in a different context. A screenshot of the lesson is shown in Figure 5.

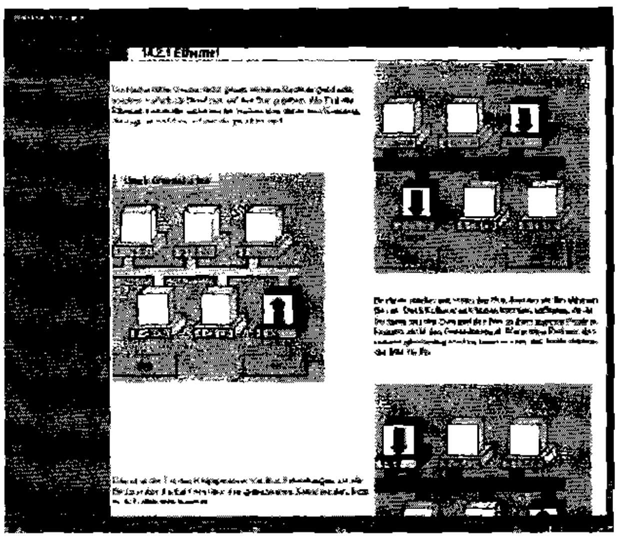


Figure 5: Screenshot of example lesson

6. 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 showed our tool implementation supprting the tagging and customization process as well an implmented example taken from the lecture we give at Darmstadt university of Technology.

We understand our article as a starting point towards the development of metadata which are suitable to deseribe 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.

7. Acknowledgements

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

8. References

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

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

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

[4] Abdulmotaleb El Saddik, Amir Ghavam and Ralf Steinmetz. Metadata for Smart Multimedia Learning Objects. In Proceedings of the fourth Australasian Computing Education Conference. ACM Press, New York, USA, December 2000.

[5] Abdulmotaleb El Saddik, Stephan Fischer, and Ralf Steinmetz. ITBeankit: An Educational Middleware Framework for Bridging Software Technology and Education. In Proceedings of EdMedia 2000, Montreal, Canada, June 2000.

[6] Abdulmotaleb El Saddik, Stephan Fischer, and Ralf Steinmetz. Reusability and Adaptability of Interactive Resources in Web-based Educational Systems. *ACM Journal of Educational Resources in Computing (JERIC)*, vol.1 issue 1, 2001.

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

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

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

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

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

[12] 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.

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

[14] R. Klein and F. Hanisch. Using a modular construction kit for the realization of an interactive computer graphics course. In Proceedings of *EdMedia97* , 1997.

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

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

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

[18] 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.

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

[20] 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.

[21] 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.