[ESFiSt00] *Abdulmotaleb El-Saddik, Stephan Fischer, Ralf Steinmetz;* **ITBeankit: An Educational Middleware Framework for Bridging Software Technolgoy and Education;** Proceedings of ED-MEDIA 2000, Montreal, Canada, Juni 2000, S.

In proceeding of ED-MEDIA 2000, Montreal, Quebec, Canada, June 2000

ITBeankit: An Educational Middleware Framework for Bridging Software Technology and Education

Abdulmotaleb El-Saddik[1], Stephan Fischer[2], Ralf Steinmetz[1,2]
1) Industrial Process and System Communications, Dept. of EE & Info. Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
2) German National Research Center for Information Technology (GMD IPSI)
Dolivostr. 15 • D-64293 Darmstadt • Germany
{*Abdulmotaleb.El-Saddik,Stephan.Fischer,Ralf.Steinmetz*}*@kom.tu-darmstadt.de*

Overview: In this paper we describe a layered approach, which allows the adaptation of the visualization content according to the user's needs using meta-data. In contrast to existing approaches, the educator can adjust both the level of explanation and the level of interactivity of an animation, thus influencing the presentation and the results of the algorithms being illustrated according to a desired level of functionality and appearance, suitable for the specific needs of the learner. Our approach is similar to a middleware approach. In this three-tier model, the lowest layer of this paradigm suits the programmer. On the middle layer, which is composed of various services common to all educational animations, such as management of data, control, interaction level and presentation, acts the educator. The user interface of the current educational visualization is the top-layer medium with which the end-user (learner) interacts.

Meta-Data within Algorithmic Animations

One way in which software evolves is to push a configuration out onto the user and to make the program as interactive as possible. An alternative way is to defer such decisions until runtime, or to push configuration decisions out into the data. Data itself become more universal and reusable when it is accompanied by meta-data. As *meta-data* we define data that describe other data, rather than aspects of the application domain itself. These data may become even more general and independent when they are integrated into the code, which may be transferred over the Internet. Using the object-oriented language Java to implement data to be described, the use of meta-data or code descriptions as objects has great advantages. Instead of using meta-data in its general term and traditional role, to create universal descriptions of objects (for example in digital library systems) these meta-data are used instead to describe properties. Runtime mechanisms for accessing, altering, adding and removing meta-data or properties at runtime should be provided by the programmer in order to integrate data with meta-data seemlessly.

Several works have dealt with classifications of meta-data, for example [Böhm et al. 94] or [InfoQuilt]. InfoQuilt classified meta-data into two categories: content-dependent and content-independent meta-data. All of the methods used to classify meta-data make use of meta-data in its traditional sense of describing data to facilitate search activities of the data (in general the documents) they describe. That is, the meta-data descriptors are associated in a fixed way with the data sets and as granular as defined initially. However, meta-data can also be seen as parameters that are passed to some object of a system in order to change its behaviour as well as providing information, which will help search engines. In order to change the behaviour (dynamic property of a resource allowing it to be integrated into a desirable context) of an object, it should be possible to change parameters on the fly. For this purpose dynamic meta-data are needed, extending the static nature of meta-data in the traditional sense.

- *Static Meta-data*
  Static meta-data are meta-data that are used to create universal and widely applicable descriptions of objects, which may not change the behavior of the described object. Examples are [IEEE-LOM] and [DC].
- *Dynamic meta-data*
  We suggest the use of the new term "dynamic meta-data", describing the possibility to adapt the content of an object according to the value given in the description and the ability to change the value of a parameter of a given object in order to change its behaviour. As an example of dynamic meta-data we examined the Ethernet protocol. In the visualization of this protocol, changing the value of the meta-data 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. Dynamic meta-data can in its turn be divided into the following two categories. We further subdivide into horizontal and vertical dynamic meta-data. *Horizontal meta-data* contain information that does not depend on the content of the algorithm to be visualized. The meta-data fields defined within this category are common for all visualization units, and thus can be applied to all animations. *Vertical meta-data* are used when the decision of changing the value of a field is up to the user. Parameter panel is an adequate representation of these meta-data.

itBeanKit

We use the itBeanKit (interactive teaching Bean Kit) to develop animation chains consisting of modular animations visualizing the steps of an algorithm. The itBeanKit comprises the following levels:

- *Programmer*
  First, the programmer identifies animation objects that visualize the steps of an algorithm. He defines the smallest entities of the algorithm and develops them as black-box software components, which may be re-used for the development of other algorithms. The programmer specifies animation actions at particular points in the algorithm chain aceording to a predefined set of meta-data. The programmer's view has been described in detail in [Elsaddik et al. 99].
- *Educator*
  As a middleware actor, 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 meta-data, which have been pre-defined by the programmer.
- *Learner*
  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.

CONCLUSION AND OUTLOOK

The itBeanKit provides an environment in which an educator, without conventional programming skills, can build an interactive visualization of an algorithm. We currently extend the itBeanKit, particularly by increasing the available choice of components. As a prototype, we implemented animations, for example "Ethernet" or "JPEG". Details can be found under the URL
http://www.kom.e-technik.tu-darmstadt.de/projects/iteach/itbeankit/applets/paradelektion/index.html

REFERENCES

[Böhm et al. 94] Böhm K. and Rakow T. (1994), "Metadata for Multimedia Documents". In Metadata for Digital Media, Special issue of SIGMOD record, 23 (4), ACM Press.

[DC], Provisional Report of the Dublin Core Subelement Working Group and http://purl.oclc.org/metadata/dublin_core/wsubelementdrafts.html

[Elsaddik et al. 99], El Saddik A,. Seeberg C., Steinacker A., Reichenberger K., Fischer S., and Steinmetz R. (1999): "A Component-based Construction Kit for Algorithmic Visualizations. In Proceedings of the Integrated Design & Process Technology "IDPT'99, June 1999, will appear in the IDPT 2000 proceedings.

[IEEE-LOM], IEEE Learning Technology Standards Committee (LTSC): Learning Object Meta-data (LOM): http://www.manta.ieee.org/p1484/

[InfoQuilt] InfoQuilt: Information Brokering for Globally Distributed heterogenous Digital Media. Large Scale Distributed Information Systems Lab, Department of Computer Science, University of Geargia. http://lsdis.cs.uga.edu/infoquilt.

[Stasko et al. 93] Stasko J., Badre A. and Lewis C. (1993), "Do Algorithm Animations Assist Learning? An Empirical Study and Analysis", ACM Interchi Conference Proceedings.