PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON INTERNET COMPUTING

IC'2001

Volume II

Co-Editors:
P. Graham, M. Maheswaran, R. Eskicioglu

Associate Editors:
H. R. Arabnia, M. Fontoura, P. H. Hanh, Y-K. Kwok, P. Langendoerfer
Y. Mun, S. K. Rahimi, A. Scime, M. H. N. Tabrizi, L. Trajkovic
A. Vakali, G. Young, J. Zadeh, B. Zhou

Las Vegas, Nevada, USA
June 25 - 28, 2001

Graph Based Visualization of Multimedia Objects and their Animation over the Internet

A. K. Bansal[1], T. Kapoor[2], and R. Pathak
Department of Computer Science
Kent State University, Kent, OH 44242, USA
{arvind, tkapoor, rpathak}@cs.kent.edu

Abstract

The improvements in image compression techniques and the rate of increase in the bandwidth will be outpaced by the rate of increase of the demand for the multimedia object transmission. In this paper, we describe the integration of mathematical graph based modeling of complex synthetic objects and the reuse of multimedia subcomponents using Single Transmission Multiple Display to reduce the transmission bandwidth requirement. This paradigm relies on image reconstruction and the reuse of a graph based model of multimedia objects at the client end and an extension of XML for graph based modeling. A grammar for STMDL — Single Transmission Multiple Display Language — and the implementation details have been provided. The model has been applied to visualize a model of a combustor engine and the animation of cartoon objects.

Keywords: Bandwidth, graphs, Internet, MPEG-7, multimedia, XML

1. Introduction

As the Internet becomes faster and more integrated, it is possible to distribute large multimedia knowledge bases over the Internet in a user transparent way. The distributed knowledge bases can be used to archive reusable libraries of multimedia objects [11]. These multimedia objects can be retrieved in real time over the Internet in a user transparent manner, and applied for many different tasks. The task could be display of animated cartoon movies, simulation and visualization of complex design objects such as combustion engine [11], cooperative design of complex objects, transmission of complex VLSI designs, design layouts, detailed geographical maps etc.

Multimedia objects and their motion are displayed at the client end as a sequence of multimedia frames integrating synchronized digitized video and sound [2]) at a fast enough rate (usually 12 to 15 video frames per second). Huge amount of data needs to be transmitted to display consecutive frames at such a rate. Four major factors determine the data rate: the size of the window, the frame rate, the quality of the picture, the transmission bandwidth.

The rate of increase in the demand to use multimedia objects over the Internet outruns the rate of increase in the bandwidth. The limitations in the available bandwidth on the web necessitates the development of alternate techniques to reduce the data transmission requirement while preserving the information.

Mathematical graphs can be used to represent the component level schematics of complex objects such as combustion engines [11], Petri nets, event based programming, and interaction between multiple objects, VLSI design, design layouts, routing details, animated cartoons etc.

Currently popular MPEG-7 specification [9] uses graphs to model conceptual graphs and to relate the actions within the scenes and event based programming. However, the current specifications do not use graphs to model complex objects and their animation at subcomponent level to reduce the bandwidth requirement. Graphs provide the ability to logically group the objects and display the multiple occurrence of invariant group of objects in the same or different frames using the previously transmitted multimedia files.

In many situations such as synthetic

[1] Corresponding author
[2] Current address: Intel Systems, Santa Clara, California, USA

animation as in cartoon movies, multimodal presentations, simulation of complex objects such as engine design, multimedia objects can be modeled by a set of multimedia subcomponents connected by a graph. The subcomponents themselves are modeled as hierarchical graphs. The multimedia objects are reconstructed at the client end by using the graph information and the image, shape, and, texture and other details of the subcomponents. By using XML to represent graphs, transmitting multimedia objects once, archiving the multimedia objects at the client end, and sending a description of the objects or motion as a sequence of operations on the graphs can notably reduce the transmission overload.

In this paper, we demonstrate the integration of graph based modeling of objects, transmission of the graph over the Internet, image reconstruction at the client end to model complex objects over the Internet to reduce bandwidth requirement. We have extended XML [17] to display graphs, and graph based visulaization of complex multimedia objects and their animation The objects are decomposed into various subcomponents. The actual picture of the subcomponents is stored in any traditional formats such as GIF, JPEG, PNG, or MPEG formats. These pictures are superimposed on the edges and nodes of the transmitted graphs to display the modified object. Various features of the graph like the size of the nodes, color of the edges/nodes, labels of the edges/ nodes, location of the nodes on the screen etc. are transmitted through the XML description. The animation has been achieved by sending sequence of XML objects representing altered subgraphs. The properties of displayed objects can also be controlled at the client end to control the rate of display and to selectively look the subcomponents at different abstract level. The overall system has been given in Figure 1.

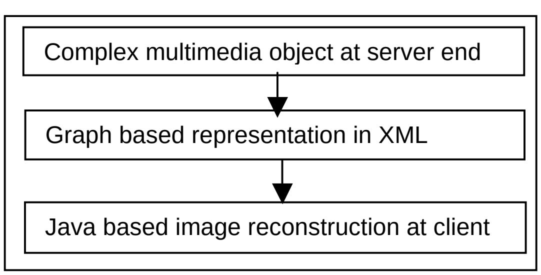

Figure 1: The overall scheme

The contributions in this paper are as follows:

1. The complex objects and their animation are modeled as graphs and operation on graphs to reuse the subcomponents at the client end resulting into reduced bandwidth requirements.
2. The use of XML for the implementation of graph based modeling facilitates integration of our techniques with MPEG-7 specifications since the data description language (DDL) for MPEG-7 is XML.
3. This implementation extends XML for graph based visualization as needed by MPEG-7.

The current system has been implemented using Java for portability, and has been interfaced with HTML for the integration of graph based modeling with other text and multimedia objects using standard browsers. The system runs on Windows, and is portable to other platforms.

The rest of the paper is organized as follows. Section 2 describes the background and basic definitions related to graph based modeling. Section 3 describes a grammar to extend XML for graph based modeling. Section 4 describes an architecture for the implementation. Section 5 describes application to the graph based modeling and animation of cartoon objects over the Internet. Section 6 describes the effects of client end buffer size and the amount of data transmitted to display animated objects. Section 7 describes other related works. The last section concludes the paper.

2. Background and definitions

This section describes an overview of the hierarchical graphs, features of XML — extensible markup language, multimedia transmission requirements, and animation.

A *hierarchical graph* (possibly directed) [8] has multiple layers of abstraction. Each level of abstraction represents a subgraph (at the lower level of abstraction) having a common attribute/function into a single node. Each node at a higher level of abstraction is a simple node or corresponds to an embedded graph at a lower level of abstraction. The edge between two nodes V_I and V_J represents one or more edges between the embedded subgraphs corresponding to the nodes V_I and V_J.

Extensible Markup Language (XML) [17] is a W3C standard document format for the data exchange over the internet. XML allows users to create their own tags, and assigns some meaning to the tags. XML documents have a hierarchical structure. An XML document consists of a root element which has multiple *subelements* and text. The *document type definition* (DTD) indicates the rules that the XML document follows. An *XML processor/parser* [18] reads XML documents and provides access to their content and structure. The XML parser takes as input an XML file and generates an XML parse tree as output. This parse tree is processed using the *document object model* (DOM) [16] — a standard API for parsers specified by the W3C.

The main parameters for multimedia transmission [3, 5, 6] over the Internet are: bandwidth, acceptable end-to-end transmission delay and delay variation (jitter), transmission reliability (data loss), distribution of traffic during time, and the importance of keeping the original timing relation during the rendering.

Many algorithms for compressing video [5] have been developed. The compression ratios of these algorithms dictate the quality of the video. The trade off among compression ratio, data rate and video quality is an important decision to transmit video over the Internet. Higher compression ratios and data rates usually require special purpose expensive hardware for decompression during playback.

3. A grammar for STMDL

An object is represented as a set of one or more embedded graphs. Each graph has zero or more attributes and one or more edges. The graph attributes could be *graph-id, default color of edges*, *default color of nodes*, *default node size* of the graph. *Graph-id* uniquely identifies a subgraph, and provides a reference from a node to the corresponding subgraph at lower level abstraction.

A multimedia object is superimposed on an edge or a node by <type> attribute where <type> is defined as an audio clip, a video clip, an image, a user-defined object, or a graph. Every node has a unique <node–id> and the coordinates to place the node at a location on the screen. The <style> attribute determines the shape of the node. The <size> attribute determines the size of the node. In the case of multiple occurrences of a node, only the first node carries the attributes, and all other nodes carry the reference-id, until the value of some attribute changes.

To simulate motion of an object a sequence of graphs modeling the object are transmitted. Only the variant multimedia objects (or operations on subgraphs representing variant parts of multimedia objects) are subsequently retransmitted; invariant part are transmitted once, archived, and retrieved from the cache using their unique id. The attributes that control motion are "superimpose", "frame_interval", "animation_ location" and "animation_number". The "superimpose" attribute allows overlay of image on the corresponding graph objects. The "frame_interval" attribute represents the time interval between the successive frames. The "animation_location" attribute specifies the location of the object within a frame and the "animation_number" is the number of frames in the frame sequence.

```
<Object>::= {<G>}+
<G> ::= ‘<’graph  {<Graph-Attribute>}*  ‘>’ {<E >}*‘
</ ‘graph ’>’
<E> ::= ‘<’edge {<Edge-Attribute>}*’>’< N> <N>
‘</’Edge’>’
<N> ::= ‘<’node  {<Node-Attribute>}* ’>’ ‘</’node’>’
<Graph-Attribute> ::= graph_id:<Number> |
edge_color:
<Color>|node_color:<Color>|size:<Number>|
superimpose: <Action> | Delay: <Number>
|animation_loc: <String> | animation_number:
<Number>
<Edge-Attributes> ::= edge_id:<Number> | color:
<Color> | label: <String> | type:<Type> | edge_dir:
<Dir>  | display: <Display>
<Node-Attributes> ::= node_id:<Number> | label:
<String>|x:<Number> | y:<Number> | type:<Type:> |
color: <Color> | size:< Number> | style: <Style> |
display: <Display>
<Dir>::= directed | undirected
<Type>::= video | image | graph | user_defined | audio
<Style> ::= solid | circle | box
<Action> ::= yes | no
<Display> ::= clickable | invisible | embedded
<Number>
<Number>::= <Digit>+
```

Figure 2. A grammar for STMDL

In the current version, all the multimedia objects in the embedded graphs are transmitted. However, multimedia objects are displayed using "display" attribute or using user control.

4. Implementation of the language

The language has been implemented using Java. The overall architecture is shown in Figure 3. A graph-based description of *a distributed multimedia object* is represented in the form of an XML file. This XML file follows the rules (given in DTD file in Figure 4) of the graph modeling grammar described in Figure 2.

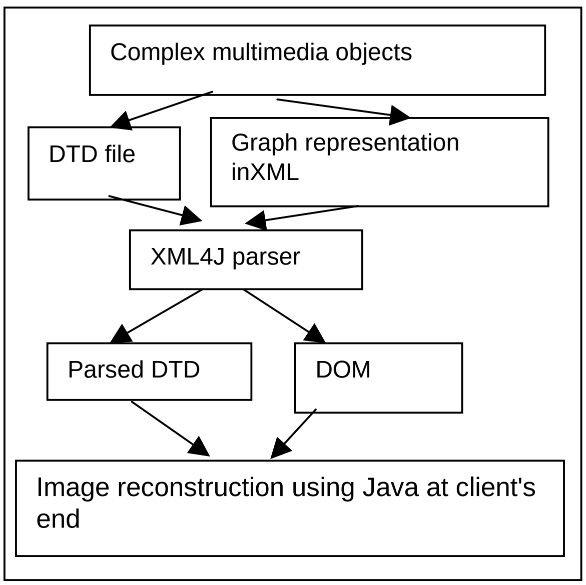

Figure 3. The overall architecture

The XML file is parsed, after verifying the conformance against the DTD, using IBM's XML parser [18] due to its capability to support DOM. DTD indicates that the graph element could be described by one or more edges, the edge element by one or more nodes and the node element by either another element or character data (denoted by the ANY content specification). The attribute lists for the graph, edge and node elements have been described. The CDATA attribute type indicates that the attribute can have character data values. Some attributes have been assigned default values, which are represented within quotes. The graph-id, node-id and edge-id are user defined attributes to uniquely identify each graph, node and edge respectively. The parser generates a logical tree. The document object returns a reference to the root of the DOM tree. Starting with the root the document is parsed using the *depth-first algorithm.*

```
<!ELEMENT graph  (edge)+ >
<!ATTLIST graph graph_id CDATA #REQUIRED
edge_color  CDATA "black"
node_color  CDATA "black" size CDATA "5"
superimpose CDATA "no"  delay CDATA "0"
animation_location CDATA ""animation_num
CDATA "0">
<!ELEMENT edge (node)+ >
<!ELEMENT node ANY>
<!ATTLIST node node_id CDATA #REQUIRED
label CDATA ""   x CDATA "0"
y CDATA "0"  color CDATA "black" style
CDATA "solid" size CDATA "5"
node_type CDATA "" image_loc CDATA ""
video_loc CDATA "" audio_loc CDATA ""
graph_id CDATA "">
<!ATTLIST edge edge_id CDATA #REQUIRED
label CDATA ""  edge_type CDATA ""
edge_dir CDATA "" color CDATA "black"
image_loc CDATA ""  video_loc CDATA ""
```

Figure 4. A DTD file for modeling

5. Applications

Complex objects are modeled using hierarchical graphs [8]. Hierarchical graphs and the corresponding subcomponents are transmitted over the Internet using XML for visualization. Figure 5 illustrates a simple graph-based model of an aircraft engine using five nodes: *environment, compressor, combustor, turbine and nozzle*. Each node represents a subcomponent. These subcomponents are modeled in more detail by expanding those nodes into sub-graphs, resulting in a set of graphs to model the entire engine. Five directed edges represent the interaction and flow of control between the physical subcomponents.

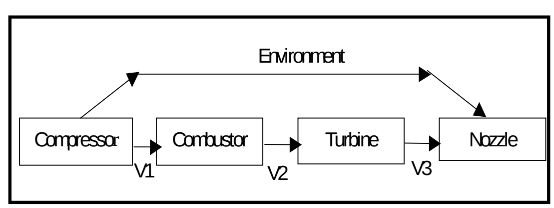

Figure 5. A graph-model of an engine

The corresponding XML file is given in Figure 6. Except the first occurrence, following occurrences of the same nodes have only reference-id. Also note that each box has a clickable MPEG animated object showing the working of the corresponding component.

5.1 Visualization of animated objects

A cartoon object is represented by a graph interconnecting different parts of the body. This description is transmitted in XML to the client side. Motion is achieved by transmitting the invariant components of the cartoon movie only once and then reconstructing the movie at the client end from the graph description.

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE graph SYSTEM "graph.dtd">
<graph graph_id="0" edge_color="black"
node_color="black">
<edge edge_id ="1" label="ENV" edge_dir="directed">
<node node_id="1" label="ENV" style="solid" size="5"
x="300" y="100" ></node>
<node node_id="2" label="compressor" style="Box"
type =  video  video_loc = C:\ compressor.mpeg display
= "clickable" size="35" x="100" y="200">  </node>
</edge>
<edge  edge_id ="2" label="V1" edge_dir="directed">
<node  node_id="2"  ></node>
<node  node_id="3" label="combustor" style="Box" type
=  video  video_loc = C:\ combustor.mpeg display =
"clickable" size="35" x="200" y="315">  </node>
</edge>
<edge edge_id ="3" label="V2" edge_dir="directed">
<node node_id="3" ></node>
<node node_id="4" label="turbine" style="Box" type =
video  video_loc = C:\ turbine.mpeg display = "clickable"
size="35" x="400" y="315">  </node>
</edge>
<edge edge_id ="4" label="V3" edge_dir="directed">
<node node_id="4" ></node>
<node node_id="5" label="nozzle" style="Box" type =
video  video_loc = C:\ nozzle.mpeg display = "clickable"
size="35" x="500" y="200">  </node>
</edge>
<edge edge_id ="5" label="ENV" edge_dir="directed">
<node node_id="5" ></node>
<node node_id="1" >  </node>
</edge>
</graph>
```

Figure 6. A representation of an engine

(i) The graph based representation provides a logical grouping of subcomponents,

(ii) Only moving subcomponents or operations on subgraphs (describing a group of moving subcomponents) needs to be transmitted to describe the motion of the subcomponents reducing the bandwidth, and

(iii) The objects can be duplicated and relocated to any part of the picture without retransmission.

A value of "yes" for the attribute "superimpose" superimposes the corresponding multimedia object on the corresponding node or an edge. The attributes "animation_loc" and "animation_num" determine the location of the first frame to be displayed and the number of frames to be displayed respectively. The "delay" attribute provides the interval (in milliseconds) between successive frame to control the speed of the cartoon animation.

This concept is demonstrated using a graph based model of *Pink Panther* and the corresponding image reconstruction by superimposing the image frames on the transmitted graph. The XML file used to generate this graph is shown in Figure 7.

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE graph SYSTEM "graph.dtd">
<graph graph_id="0" superimpose="yes"
animation_loc="c:\ complete" delay="100"
animation_num="3">
<edge edge_id="1" color="pink" type="image"
image_loc="c:\ body.jpg">
<node node_id="1" size="4" color="pink" x="300"
y="60" type="image" image_loc= "c:\ head1.jpg">
</node>
<node node_id="2"  size="2"  x="300" y="200" " type =
"image" image_loc = c:\torso.jpg > </node>
</edge>
<edge edge_id="2" color="pink" type="image"
image_loc="c:\ leg_1.jpg">
<node node_id="2" > </node>
<node node_id="3"  size="1"  x="325" y="150" type =
"image" image_loc = "c:\feet_1.jpg> </node>
</edge>
<edge edge_id="3" color="pink" type="image"
image_loc="c:\ leg_2.jpg">
<node node_id="2" > </node>
<node node_id="4" size="1" x="275" y="140"
image_loc = "c:\feet_2.jpg> </node>
</edge>
```

Figure 7. A representation of a cartoon

The graph consists of 25 edges and 27 nodes, each of which has a unique id number. Different edges and nodes of the graph are associated with different body parts of the *Pink Panther* object. The complete object is recreated by displaying all the subcomponents. Due to space limitation, we have reduced the example to a simple one using 4 nodes and 3 edges.

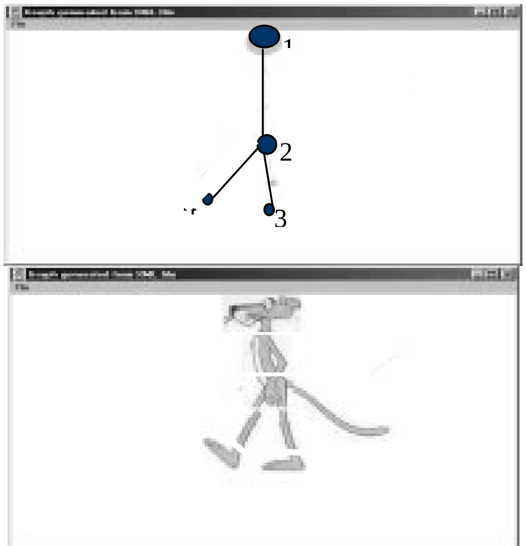

Figure 8. A representation of a cartoon object

6. Performance evaluation

We simulated the Internet transmission, and used a buffer at the client end. The overall behavior of the buffer size compared to the amount of data transmission summarized in Figure 9.

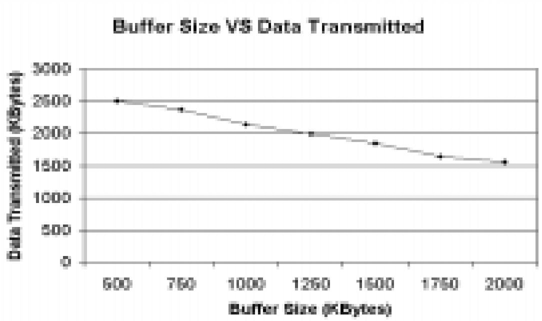

Figure 9. Buffer size vs. data transmitted

The X-axis shows the available buffer size, and the Y-axis shows the amount of data which was transmitted (or retransmitted due to the lack of available buffer space) to ensure the display of animated multimedia objects. As the buffer storage is increased, the archived objects can be retrieved from the cache, and the server does not have to retransmit the objects.

7. Related works

Currently popular tools for graph-based visualization [1, 4, 10] support animation of algorithms. However, none of these tools have any provision for Internet based modeling and animation of the complex objects over the Internet. The system also does not provide support for XML — a de facto standard for web based markup scripting and MPEG-7 [9] DDL.

MPEG-7 specifications and simulations use graphs to model conceptual graphs, event based programming, and scene decomposition. However, MPEG-7 specifications do not use graph based modeling of complex objects at subcomponent level to logically group and reuse the subcomponents. Instead MPEG-7 uses grouping at pixel level, mosaic level, spatial segment level. While grouping at pixel level and mosaic level can provide much detailed description of objects or scenes involving non-rigid objects, logical grouping based on subcomponents, reuses the subcomponents effectively when the same object or subcomponent is occurs randomly in sequence of frames. The graph based modeling technique described in this paper, when integrated with MPEG-7 specifications, will enhance the reuse of the subcomponents of repetitive scenes and synthetic objects, and help reduce the bandwidth requirement significantly.

VRML — Virtual Reality Modeling Language — based systems also uses client end computability to reduce the bandwidth requirement. However, our scheme uses real images and archival of real invariant images to reduce the bandwidth requirement. In future, we intend to integrate our work with VRML generated figures and realistic rendering and shading software to get a realistic scene reconstruction and rendering with even lesser bandwidth.

8. Conclusions and future work

In this paper, we have integrated graph based visualization, modeling of complex objects using hierarchical graphs, STMD (Single Transmission Multiple Display) and image reconstruction at client's end to reduce the transmission bandwidth requirement on the Internet. We have described a markup language to display graphs by extending XML. The markup language has been embedded inside HTML for a browser based display. The markup language has been applied to visualize the complex objects and their modeling over the Internet and to visualize animated cartoon movies. A hierarchical graph based modeling can not currently model complex scene due to its inability to capture relationship between objects and events. However, the STMDL supports a general notion of graph since each node and each edge has its own reference-id. A scene can be modeled as a group of hierarchical graphs which are connected through relations and events. However, a detailed description of modeling of scenes is outside the scope of this paper. Currently, we are extending this graph based modeling scheme for 3-D jitter free modeling of complex multimedia scenes where multiple objects interact. We are also integrating this work with demand driven visualization where transmitted data is archived in slow storage and retrieved at user demand by looking at the XML description.

References

[1] S. Bridgeman, A. Garg, and R. Tamassia, "A Graph Drawing and Translation Service on the WWW," Graph Drawing '96, Berkley, CA, USA, Springer Verlag, LNCS 1190

[2] ESCAtech media inc, *Computer Sound Formats*, http://www.escatech.com/soundpage/sound.html

[3] L. Fan, P. Cao, W. Lin, and Q. Jacobson, "Web Prefetching Between Low-Bandwidth Clients and Proxies: Potential and Performance." ACM SIGMETRICS 1999

[4] M. Fröhlich, M. Werner: "Demonstration of the interactive Graph Visualization System daVinci, *Proceedings of DIMACS Workshop on Graph Drawing*, (1994), Springer Verlag; LNCS. 894; 266 -269

[5] B. Furht, R. Westwater, and J. Ice, *Multimedia Broadcasting over the Internet: Part II-Video compression*, IEEE Multimedia, (1999), 6:1, 85-89

[6] P. A. Gloor and F. Makedon, "Obstacles in Web Multimedia Publishing," *Multimedia Tools and Applications*, (1998) 6, 171–180

[7] I. Herman, G. Melançon, and M. S. Marshall, "Graph Visualisation and Navigation in InformationVisualisation: a Survey," *IEEE transactions on Visualization and Computer Graphics*, (2000) 6:1, 24-43,

[8] Y. Ioannis and I. G. Tollis , "Graph Drawing and Information Visualization," *ACM Computing Survey*, (1996) 28

[9] J. M. Martinez, " Overview of the MPEG-7 Standard,", ISO/IEC JTC1/SC29/WG11 N3752, http://www.cselt .org/mpeg

[10] P. J. Rodgers, "A Graph Rewriting Programming Language for Graph Drawing," Proceedings of the 14th IEEE Symposium on Visual Languages and Computing, Halifax, Nova Scotia, Canada, (1998), 32 – 39

[11] S.W. Ryan, A. K. Bansal,, and T. Kapoor, "A Graphical Web-Based Multimedia Tool for Distributed Modeling over the Internet," *Proceedings of the International Conference of Tools with Artificial Intelligence*, Vancouver, Canada, 2000, pp. 140 - 146

[12] G. Sander, M. Alt, C. Ferdinand, and R. Wilhelm, "CLaX, A Visualized Compiler," *Proceedings of the Symposium on Graph Drawing, 1995, Lecture Notes in Computer Science* 1027, 459-462, Springer Verlag, 1996,

[13] W. D. Schwaderer, *"Digital Imaging in C and the World Wide Web,"* Wordware Publishing, Inc.

[14] F. Siqueira, "A Framework for Distributed Multimedia Applications based on CORBA and Integrated Services Networks,"http:// www.cs.tcd.ie /Frank.Siqueira/PhD-Project/ Requirements.html, Ph. D. Thesis, Trinity College, Dublin, Ireland

[15] VRML Consortium, The Virtual Reality Modeling Language, http://www.vrml.org/ technicalinfo/ specifications /vrml97/index.htm

[16] World Wide Web Consortium, Document Object Model (DOM) 1.0, http://www.w3.org/TR/REC-DOM-Level-1/

[17] World Wide Web Consortium, "*Extensible Markup Language (XML) 1.0*", http://www.w3.org/TR/1998 /REC-xml-19980210

[18] *XML parser for Java*, http://www.alphaWorks. ibm.com/tech/XML4J