Wireless Internet for The Mobile Enterprise Consortium http://winmec.ucla.edu/

TECHNICAL REPORT/WHITE PAPER
UCLA-WINMEC-2005-201-SVG-EXCHANGE

SUBMITTED TO ACM TRANSACTIONS ON MULTIMEDIA COMPUTING, COMMUNICATIONS AND APPLICATIONS (TOMCCAP)

Scalable Vector Graphics (SVG) based multi-level graphics representation for engineering rich-content exchange in mobile computing environment

Xiaoyong Su, B.S. Prabhu, Chi-Cheng Chu, Rajit Gadh
UCLA – Wireless Internet for Mobile Enterprise Consortium (WINMEC)

This research investigates the use Scalable Vector Graphics (SVG) to support 2D multimedia-based collaboration for mobile users and workers for such potential activities as gaming, advertising, medical analysis, sales, out of office professional services, logistics, repair and maintenance over wireless enabled mobile devices. The challenge here is to enable rich-content delivery over wireless wide area networks which are typically bandwidth constrained and to mobile devices which are computationally constrained. In our research the guiding application chosen is engineering field service and the approach entails transforming graphical engineering content, drawing, schematics, plans, circuits, maps, etc., into XML based representation, which maintains a hierarchic scheme for related information of assemblies, sub-assemblies, part drawings, bill-of-materials, etc. The hierarchic content is delivered in a granular format to the handheld devices over wireless Internet based on user request - thereby addressing both the problems of limited bandwidth and computing power on the device. The framework can handle “level of details” and “region/point of interest”, redlining, collaborative discussion in the field, remote data enrichment, content visualization, session management, and other activities germane to field service needs. The paper also discusses the performance and experience in handling some industrial engineering content.

Categories and Subject Descriptors:
General Terms: Mobile Computing, Engineering Content Exchange, Architecture, Framework
Additional Key Words and Phrases: Mobility, mobile computing, heterogeneous computing environment, Scalable Vector Graphics, multi-level division, graphics decomposition and composition, on-demand content delivery, content discovery, engineering graphics, rich-content, multimedia

---

1. INTRODUCTION

The adoption of mobile computing and wireless communication technologies are changing the traditional business operations and practices in enterprises. A variety of mobile-working models are currently under consideration by research, development and user community both in industry and academia, as mobile ubiquitous computing is perceived to provide the industry with an avenue to do business in a better way. For example, the chip giant Intel announced Personal Client Architecture in 2001 to support high-performance applications on ultra low power mobile devices with integrated voice and Internet capability. Similarly software giant Microsoft is working on a focused mobile computing strategy. Lots of other organizations such as World Wide Web Consortium (W3C) and Open Mobile Alliance (OMA) are channelizing their research and development efforts in standardizing mobile content organization, adaptation, security, representation and delivery. .

Today, latest mobile devices such as cell phones, smart phones, and handhelds, commonly support rich multimedia content, increasing their appeal and enabling them to be used as personal digital assistant, entertainer, organizer and communication tool. As more and more device vendors support powerful processors such as XScale/wireless MMX chips and Open Multimedia Application Platform, the new generation mobile devices will become a serious computing platform for both enterprises and consumers running sophisticated applications and services.

Among obvious technology advances in mobile computing, there are still a number of issues which need to improve. Foremost issue being the conflict arising out of the need to support ubiquitous mobility and better performance, on account of usually small screens, limited power and computation capability, and constraints on user-input methods. On the other hand, wireless communications is fraught with problems of low bandwidth, intermittent connection, and signal interference. All these factors pose a formidable challenge to mobile computing environments (MCE) in general and more so when multimedia content is part of the mix [ROSENBAUM, et al, 2004].

Graphics content in a variety of formats can be gainfully utilized in mobile environments for different requirements. 2D vector graphics can be intelligently used for graphics based design; location-based services such as traffic and weather reports; mapping, positioning and navigating; multimedia messaging; animation and interactive graphics; and entertainment. To support these applications or services, usually it will be required to transmit and handle content over heterogeneous networking - network service with different capabilities from multiple carriers; and computing environment – number of mobile devices with different capabilities and features.

1.1 Typical challenges of mobile multimedia delivery

The problems generally don't present when transmitting small size 2D vector graphics files. But when transmitting large sized files, say for example a 2MB DXF file to a GPRS cell phone the problems manifest in more ways than one. Ideally, the maximum bandwidth of the GPRS connection is 171kbps. At this rate the total download time Td will be 2000*8/384 = 93.6s. However, the actual download time will exceed the theoretical value because usually the bandwidth available will be 30-60 kbps based on the device and the implementation of the service by the carrier. This situation potentially will lead to service interruptions and erroneous downloads leading to higher costs. In case the file gets downloaded without errors; the parsing and rendering processes on the current small devices are unbearable.

Generally graphics content is represented in a single monolithic file, be it a CAD design, vector map, etc. To utilize such graphics content, the single file based representation will have to be transmitted over networks; will have to be downloaded in its entirety into device memory and rendered. This mechanism of graphics content delivery is a legacy of the desktop environment, where fast network connection is a norm and computing capability of the device is not an issue. But, this mechanism fails miserably in a MCE.

Table 1 shows the theoretical data of transmitting graphics files of different sizes over different networks to give a perspective of performance.

Table 1. the downloading time comparison (Bandwidths of GPRS and 802.11b are assumed consider various factors)

	20KB	200KB	2MB
GPRS (36Kbps)	4.4s	44s	440s
802.11b (2MBps)	0.08s	0.8s	8s
100M Fast Ethernet	~0	~0	0.16s

Similarly for the sake of comparison, Table 2 shows the time required to render JPEG images (raster graphic content) of different sizes on a mobile device and a desktop system.

Table 2. Time required comparison of rendering JPEG file on mobile device and desktop system

	17KB	165KB	622KB
Mobile Device (iPaq5450, XScale 400MHz, 64M RAM)	<1s	~3s	~9s
Desktop computer (Compaq Evo, 2.4GHz, 512MB RAM)	~0	~0	<1s

Thus for successful mobile multimedia delivery, the framework will have to take a comprehensive recourse to content representation, progressive delivery and provide intelligent user interaction.

In this paper we report one such novel framework for engineering field service applications where rich-content is predominantly 2D (drawings, schematics, layouts, diagrams, etc.). Our attempt is to create a Multi-Level 2D vector graphics modeling and representation concept, and a progressive delivery method using standards based Scalable Vector Graphics (SVG) [http://www.w3.org/Graphics/SVG/] and eXtensible Markup Language (XML) [http://www.w3.org/XML/] to try to mitigate the twin principal problems of mobile computing viz., low bandwidth and computing power. We are also confident the system can be successfully extended to such activities as gaming, advertising, and medical analysis where 2D graphical information is used gainfully.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 explains the system overview of our layered 2D vector graphics and delivery concept. Details of layered 2D vector graphics representation scheme forms section 4. Section 5 describes the details of content delivery and interactive operation. Our prototype system and its working are explained in section 6. Finally, we draw conclusions on the research in section 7.

2. RELATED WORK

In this section prior related work on multimedia in general, 2D multimedia modeling and representation, visualization and standards will be discussed in brief.

2.1 Multimedia in enterprise

Multimedia would enable strong first impression, mitigating ambiguity and influencing better decision making [LIM, et al, 2000][GROSKY, et al, 1997]. The importance of first impression is more pronounced in online activities and transactions as content needs to be presented to the user in such a way that sufficient information is conveyed in a self explanatory manner, inviting the user to explore it by interacting with it, imparting a very high degree of 'quality of experience' as highlighted by Ramesh Jain [2004]. With intelligent organization of multimedia content it is possible to add syntax and semantics which can be represented as metadata of the content, which in turn aids content management and location[JAIN, 2003][GUNASEKARAN, et al, 1999].

Our work subscribes to these ideas - the choice of content representation standards; the proposed hierarchic representation schema, intelligent delivery and interaction reflect this thinking, and demonstrates a better quality of experience to mobile multimedia users.

2.2 2D Vector Graphics formats

Standard 2D vector data exchange formats have been widely used in multimedia 2D graphics and engineering data representation. The existed vector data exchange formats can be divided into two categories based on their end-use: vector format for engineering content and vector format for multimedia web content. The widely used vector formats for engineering include DXF, CGM, DRW, DWG, PCL, HPGL, PCT, and SHP. Popular vector formats for multimedia web content include WebCGM, CMX, WMF, SVG, and VWPG.

Drawing Interchange Format (DXF) [http://www.autodesk.com/techpubs/autocad/dxf/] was first introduced by AutoDesk in AutoCAD, a popular CAD software, and is one of the most widely used CAD formats. DXF format is a tagged data representation format. Each data element in the file is preceded by a *group code*. A group code's value indicates what type of data element follows. This value also indicates the meaning of a data element for a given object (or record) type. By using group code, the DXF file is organized into sections. The sections in a DXF file include HEADER, CLASSES, TABLES, BLOCKS, ENTITIES, OBJECTS and THUMBNAILIMAGE. The entities in ENTITIES sections contain various 2D and 3D graphics primitives of the drawing or model. The DXF file can represent almost any CAD drawing using those entities and can connect a group of entities together to form a block (such as windows, doors, etc. in an architectural drawing).

Scalable Vector Graphics (SVG) [LAU, et al, 2003], which has been developed by World Wide Web Consortium (W3C), is relatively a new standard for two-dimensional vector graphics. Since it is an XML-based graphics description language, it inherits the features of XML such as portability and interoperability. SVG has three types of graphics objects: vector graphics, raster graphics (images, symbols and makers) and text. Graphics objects can be grouped, styled and transformed into reusable components. SVG supports rich graphics primitives such as line, polyline, polygon, path, ellipse, circle, rectangle and special effect such as filter, gradient filling, alpha masks, etc. Another important feature that SVG supports is linking.

SVG can be used for design, GIS and mapping, embedded systems. location-based services such as traffic and weather reports, mapping and positioning, that's useful for navigating, animated picture messaging, multimedia messaging, animation and interactive graphics; entertainment; eCommerce and user interfaces.

Currently, there are three versions of SVG specification: SVG 1.2, SVG tiny for cellphones and SVG basic for PDAs. SVG tiny and SVG basic together are called Mobile SVG.

2.3 Graphics visualization

There are few research works reported related to handling of big 2D graphics files in a MCE. Most of these are for visualizing large data sets. Usually, hierarchical representation [ABELLO AND KORN, 2002] [FLORIANI AND PUPPO, 1995], server based computing [LEIGH AND BAILEY, 1999] [JING, et al, 1999] and tree data structure are most commonly used methods.

Abello J., et al, [2002] present a hierarchic representation for massive multi-digraphs. They propose a method to transform an arbitrary graph into hierarchies by building a hierarchy of multi-digraph layers on top of the input multi-graph. Each layer consists of sets of vertices of previous level. By using the hierarchy structure, they can control the display space efficiently. Jason Leigh and Stuart Bailey [1999] propose a methodology for supporting collaborative exploratory analysis of massive data set in tele-immersive environments. In this methodology, a remote rendering server will render the viewpoints as a sequence of stereoscopic images or animations. Then the images and animations will be compressed and streamed to the client. Paula Frederick et al, [1999] present their study on improving the performance of visualization of large figures. They use a persistent structure that is composed of an R-tree [SELLIS, et al, 1987] [GUTTMAN, 1984] and V-Tree [MEDIANO, et al, 1994] for storing and retrieving 2D spatial data. Bin Pham and On Wong [2004] have analyzed the capabilities and limitations of current handheld devices and discuss the important issues to be considered such as data organization, management, communication, input methods and user interfaces.

Among these methods, hierarchic representation has big advantage over the other two methods. But most research on this issue don't mention about how to transmit the content progressively. Server based solutions have to maintain a state for each session and need lots of computational resource on the server side. It is not suitable for a massively distributed computing environment. The tree structure representation method solves the storing and retrieving issue in massive data sets, but it is still hard for mobile devices to consume large data in these cases.

2.4 Using SVG standard for engineering data visualization

As a new vector graphics standard, SVG has been widely used for 2D graphics data representation in various fields. Some researchers have used it as visualization language for different type of scientific data.

Andres Baravalle, et al, [2003] use SVG and XSL to visualize dynamically changing data. Yi-Hong Chang, et al, [2002] use SVG to visualize census data online. C.T. Lewis, et al, [2002] use SVG to visualize medical data. Other applications include GIS [BARU, et al, 2001], human navigation [KOBAYASHI, et al, 2003] etc.

Sangmi Lee, et al., [2002] report an SVG based collaborative system Garnet for distance-education running on desktops and PDAs. The architecture of Garnet is based on event brokering system. They claim that SVG provides better graphics and document interactivity. They highlight the fact of the possibility of separation of presentation and content layers using SVG, providing flexible viewpoints to the users.

3. SYSTEM OVERVIEW

As mentioned earlier 2D graphics usually is represented in a single file containing lots of information. As a result, to use this content, no matter what format it is in, it is required to be transmitted and read as a single file into the device memory, subject to the bandwidth and computational capability of the MCE. We present a multi-level 2D graphics model in this paper in an attempt to solve these problems. For this we do not intend to create another new format but use existing standard data exchange formats only with a new representation framework. The underlying key concept of our work is to represent content into intelligent granular levels and to deliver the same on demand. By using intelligent interpretation of engineering drawing [PRABHU AND PANDE, 1999], 2D engineering can be parsed and decomposed into multilevel representation.

3.1 Multi-Level Subdivision

Based on the properties or syntax of 2D graphics content, we can extract the primitives that have same or similar properties and separate them into files. Here we introduce a three level graphics representation mechanism. Firstly common properties and attributes such as name space description, coordinate system, etc., are extracted and they constitute the top level graphics domain description file or the 'Root'. At the second step, graphics views, profiles and graphics objects are grouped forming Level 1. Groups of low-level primitives form the high-level graphics object in this level. Level 2 constitutes the individual graphic primitives. This process is called as multi-levels division.

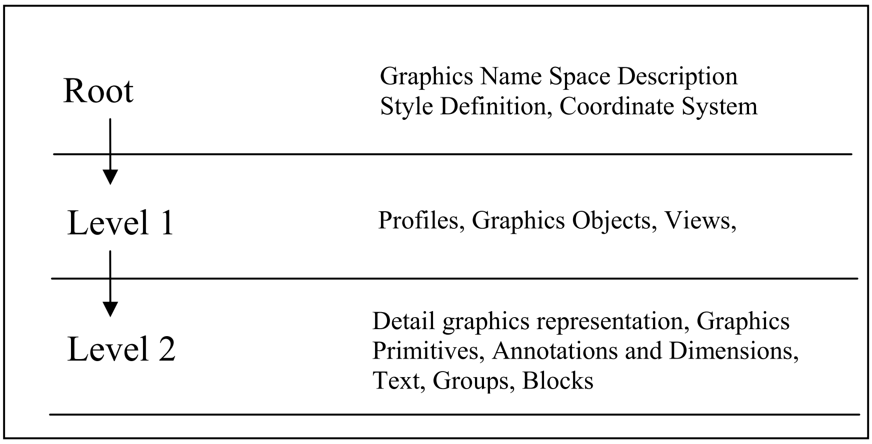


Fig. 1. Multi-Level Division

Figure 1 shows a typical multi-levels division. Root, a single description file, is the entry point of graphics content. This file contains information including graphics description (Vendor, Reference, Name Space, etc) and style definition that will be used for all the primitives in this name space. Level 1 consists of profiles, graphics objects (such as image, symbol, marker and graphics components etc), Views (Multiple views are possible to be used to describe a graphics domain). In this level multiple files will be used to represent all the information. Level 2 consists of detail graphics primitives, Annotations, Dimensions, Groups, Blocks, and Text. Criteria for subdivision can be based on primitive type, region of interest, pre-defined templates, and even user selection.

When the graphics content is broken into different levels, it should be done in such a way that the level of detail achieved is manageable as well as the granularity will suit the MCE. Too much detail will cause too many small size files which would need extra bandwidth. This would provide obvious benefits such as reducing the problems attributed to service interruptions, facilitate short duration transmissions, and only deliver content of interest.

3.2 Tree structure representation

In this section, we will discuss how to subdivide single 2D graphics content in detail.

If GN represents a graphics domain, $LN1_i$ represents an object in Level1, $LN2_{ij}$ represents the sub-components of object $LN1_i$ and symbol $\oplus$ represents content merging, we can establish the following relationship:

$$GN = \sum LN1_i = LN1_0 \oplus LN1_1 \oplus \ldots\ldots \oplus LN1_n. \text{ where } i \in (0, n)$$

$$LN1_i = \sum LN2_{ij} = LN2_{i0} \oplus LN2_{i1} \oplus \ldots\ldots \oplus LN2_{im.} \text{ where } j \in (0, m)$$

If we represent this relationship in a tree structure, it can be shown as in figure 2. In this tree structure, each node represents a single file or directory. We use a graphics domain which has multi-level tree structure to organize these files. The graphics domain looks like name space in JAVA. For example, if the file name of Level1 node $LN1_0$ is 'PartProfile.svg' (profile of a part in one of the views) and Level2 node $LN2_{01}$ is 'hatch.svg' (hatching of the profile of the part), then the full lookup index of this branch of the tree will be *YOUR_ORGANIZATION.GN.LN1₀[PartProfile.svg].LN2₀₁[hatch.svg]*. These nodes can be dispersed over a number of machines in an enterprise network. By using such a method, graphics content can be easily located in a distributed environment.

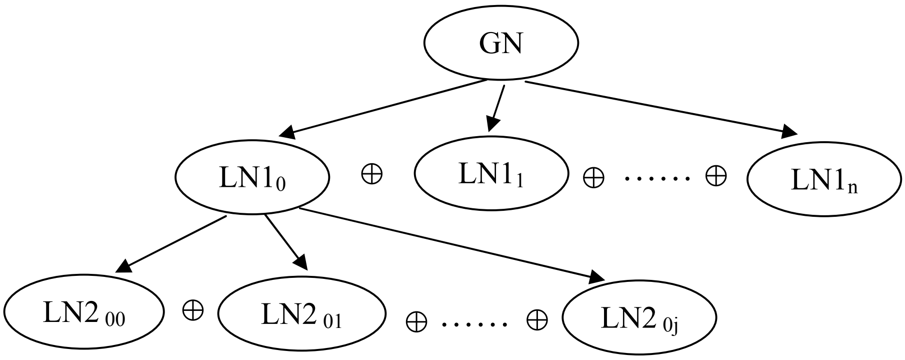


Fig. 2. Tree structure of the multi-level division

3.3 An instance of Multi-Level Division

Figure 3 shows an engineering drawing containing Multi-Level information. It has three views, textual annotation, drawing information, dimensions, etc., and each of these can be separated into different nodes (files) containing semantically related information.

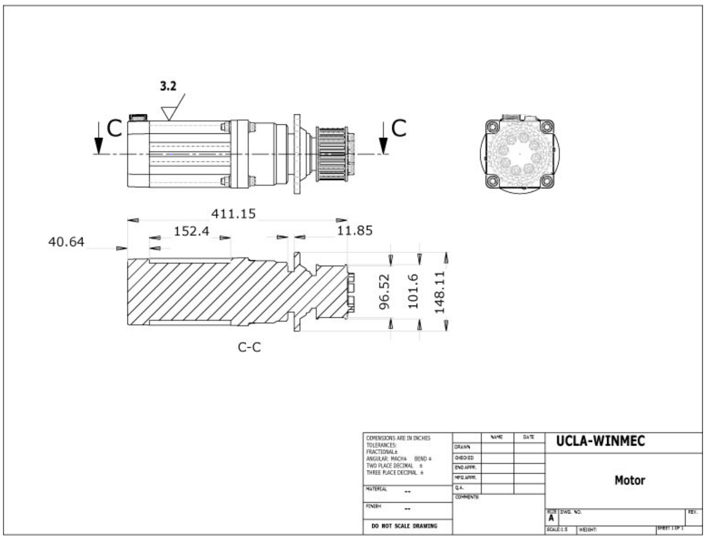


Fig. 3. An engineering drawing with variety of information

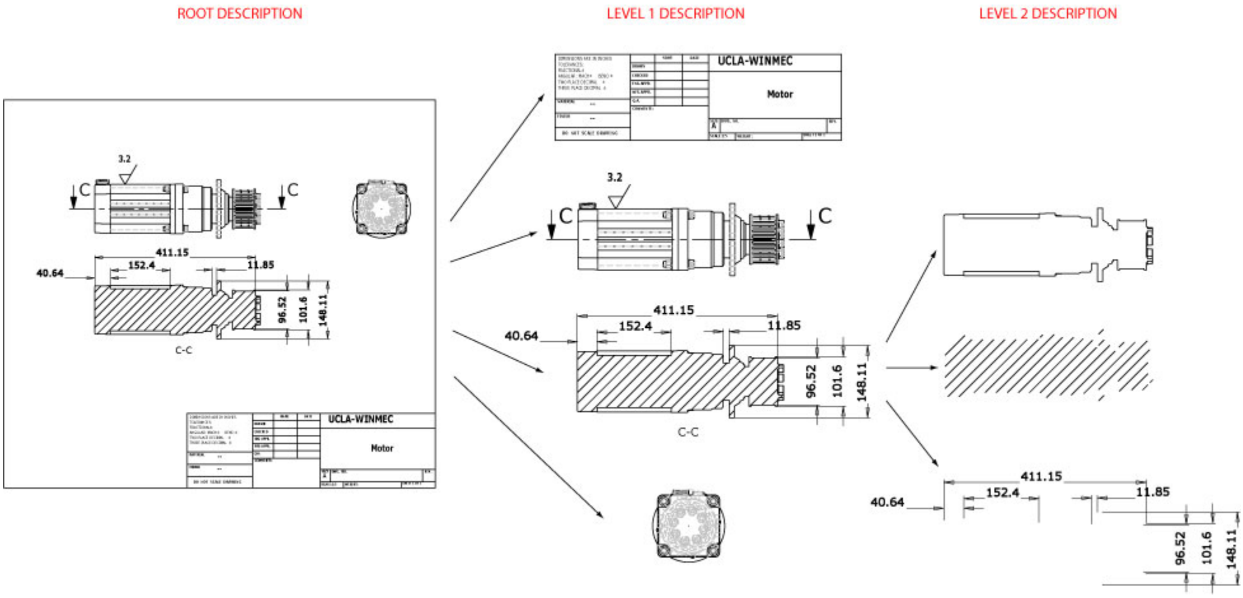


Fig. 4. Sub-division sequence and layout

Firstly, the common properties such as line style, line width, font face and size, etc., are extracted. At the same time, the first level division will be conducted. The graphics content will be divided into sheet, main view, left view and top view.

In the process, common properties, along with the general description of this drawing such as vendor, language etc, will be used to form the root and it will constitute the top level description file. Starting from this file, it is possible to transverse the entire graphics content.

In the second stage, the second level division is carried out. In the above example, top view has been divided into profile, hatch and dimension.

If we reverse the whole process, we can use the divided content to regenerate the original drawing. Figure 4 shows the subdivision sequences of this engineering drawing.

4. MULTI-LEVEL 2D GRAPHICS REPRESENTED BY SVG

4.1 Short Description on SVG

Scalable Vector Graphics (SVG), which is based on eXtensible Markup Language (XML), is an open 2D vector/raster mixed graphics standard. It has a well defined file format and a programming API framework. SVG has already gained wide support from industry. A number of free SVG viewers, editors, converters and SVG Development Toolkits are available from a number of companies. The following list comprises of the different features that are supported by SVG:

1. **Graphics objects -** supports three types of graphics objects: vector graphics shapes, images and text. Vector graphics primitives are paths, rectangles, circles, ellipses, lines, polylines and polygons. The properties of graphics primitives can be either set by Cascade Style Sheet (CSS) or individual setting. Coordinate Systems, Transformations are also supported in SVG.
2. **Effect of graphics -** supports special graphics effects including Painting, Gradients, Patterns, Clipping, Masking, Compositing, Filtering, and Animation.
3. **Document structure -** content is organized into Document Object Model (DOM), which is a tree structure containing objects.
4. **Special Functionality** - supports linking, interactivity and record events. These functionalities are extremely useful in an interactive collaborative environment.
5. **Other features -** features that SVG supports include metadata, scripting, multimedia contents, etc.

SVG uses same syntax description as in XML. Elements of a typical graphics content can be described by a sequence of drawing elements of SVG. The following example gives us an idea of content organization in SVG. It represents a non-filled rectangle and a circle with some line style and default color.

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-flat-20030114.dtd">
<svg  width="300" height="225">

   <title>Sample SVG graphic</title>

   <rect x="40" y="40" height="100" width="100" fill="none" stroke="black"
                stroke-width="1" />
   <circle cx="200" cy="150" r="40"  style="fill:red;"/>
   <text x='85' y='20' fill='blue'>Greetings from WINMEC ! </text>

</svg>
```

As we can see from this example, besides the graphics description sentences, there are some other descriptions at the beginning of the file. These sentences are necessary to describe document type, styles and other useful information, but they increase the file size. So, in our design, we try to minimize these descriptions to only the essential in the graphics domain without violating the requirements of the standard thereby providing maximum portability. In the following sections, we will explain how to use SVG to dissect the above engineering drawing example.

4.2 Using SVG to describe the Top Level 2D graphics

As we described in section 3, 2D graphics content can be divided into several levels. The top level, called Root, is the entry point of the 2D graphics. It will contain information of the system such as vendor, time, name space, etc., and style definition. All this information applies to the whole graphics domain and working space. The following are examples of how to build a Root of 2D graphics domain. At this level, general information is gathered and level 1 structure has been established.

```
<! --------- SVG announcement ------------->
<?xml version="1.0" standalone="yes"?>
<root xmlns:Project1 ="http://www.wireless.edu/2004/Project1" ></root>
<svg width="600.0" height="391.429" viewBox="0 0 1050.000 685.000" version="1.1"
      xmlns="http://www.w3.org/2000/svg">

     <! ----------- Descriptions ----------->
     <title>MotorDrawing</title>
     <!—parent is important to establish a two way relationship between each file -->
     <parent>self</parent>
     <desc>
     <Vendor>WINMEC</Vendor>
          <ProductNo>PN123.342<ProductNo>
          <Version>Version 1.0001</Version>
          <Time> JAN-04-2002, 21:05 </Time>
     </desc>

      <! ----------- Styles ----------->
     <style type=”text/css”>
     ……..
    </style>

  <! ----------- graphics ----------->
  <g>
    <image xlink:href=Sheet.svg />
    <image xlink:href=TopView.svg />
    <image xlink:href=RightView.svg/>
    <image xlink:href=SectionView.svg/>
   </g>
</svg>
```

There are four parts in this description file. The first part announces a Root SVG file and defines the namespaces. The second part describes the general information of the graphics content layout. Usually, this part won't be rendered to the graphics screen, but is used in rendering the lower level objects, but this information can be display if required. The third part is cascade style sheet definition, which defines all the styles that will be used for rendering the entire graphics domain. The fourth part is the graphics structure, which points to files belonging to the next level, in the form of 'link' tags of SVG. Thus, the motor drawing is now divided into four separate files.

4.3 Using SVG to describe Level 1 and Level 2 graphics

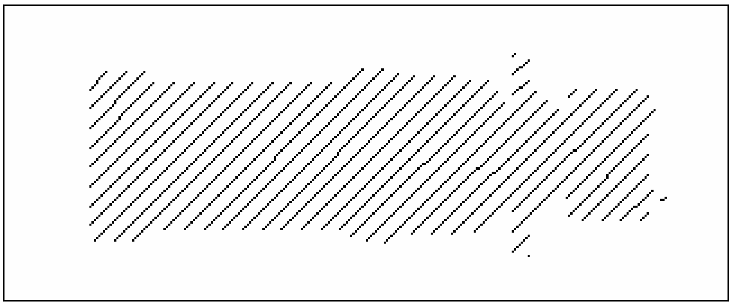
In Level 1, we need to announce the SVG parameters and describe the graphics structure. Let's use section view as an example. It is written as follows.

```
<?xml version="1.0" standalone="yes"?>
<svg width="600.0" height="391.429" viewBox="0 0 1050.000 685.000" version="1.1"
     xmlns="http://www.w3.org/2000/svg">
    <title>SectionView</title>
    <parent>MotorDrawing</parent>
 <g>
   <image xlink:href=hatch.svg/>
   <image xlink:href=profile.svg/>
   <image xlink:href=dimension.svg/>
 </g>
</svg>
```

In these levels the unique feature of SVG standard of inheriting properties is utilized to maintain contextual relationship between levels. In general if the content is displayed in standalone format, then it uses the current SVG parameters. But, if the current content merges into parent level, then it uses the parent's parameters. The graphics structure of Level 1 uses the properties of the Root level in our representation. Here again 'links' are used to point to the next level graphics sub-objects or primitives. Section view now has been divided into three files.

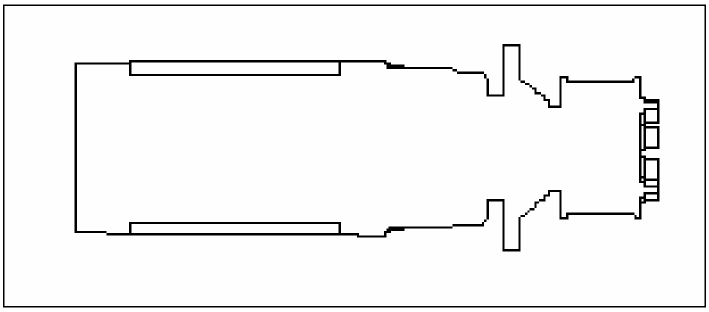
Level 2 represents actual graphics details. It mainly consists of graphics primitives, which can be grouped and transformed to form the desired graphics content. Following example shows how it is organized. For brevity only sections of the SVG file is presented here. The complete SVG file structure can be found at [http://wireless.ucla.edu/].

Hatch.svg



```
<?xml version="1.0" standalone="no"?>
<svg  width="600.0" height="391.429" viewBox="0.0 0.0 1050.000 685.000">
   <title>hatch</title>
   <parent>SectionView</parent>
  <g id="group1" fill="none" >
<path d="M270.87 342.7L263.46 350.12M279.83 …… " fill="none"stroke="black" stroke-width="0.2"/>
……
   </g>
</svg>
```

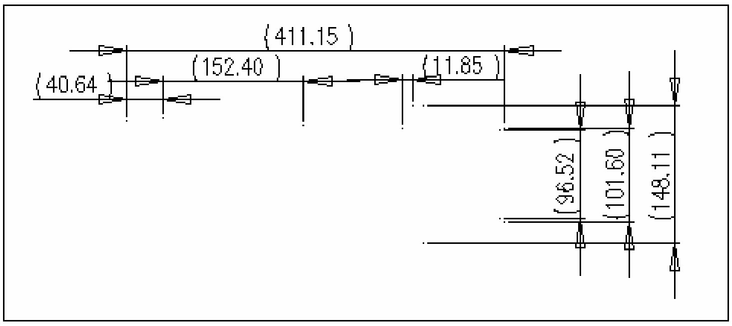
Profile.svg



```
<?xml version="1.0" standalone="no"?>
<svg width="600.0" height="391.429" viewBox="0.0 0.0 1050.000 685.000">
   <title>hatch</title>
   <parent>SectionView</parent>
  <g id="group1" fill="none" >
<path d="M515.28 456.23L516.52 456.23M516.52 444.57L515…… "
fill="none"stroke="black" stroke-width="0.2"/>
……
```

```
  </g>
</svg>
```

Dimension.svg




```
<?xml version="1.0" standalone="no"?>
<svg width="600.0" height="391.429" viewBox="0.0 0.0 1050.000 685.000">
  <title>hatch</title>
  <parent>SectionView</parent>
 <g id="group1" fill="none" >
<path d="M558.94 398.43L559.96 398.05 " fill="none"stroke="black" stroke-width="0.2"/>
……
  </g>
</svg>
```

This process decomposes a large single file into multiple small files. In this case, the original file is 233KB. By doing ‘level1’ subdivision, it has been divided into rightview.svg, which is 20KB; mainview.svg, which is 27KB; sheet.svg, which is 28KB; and sectionview.svg, which is 20KB. And going to ‘level2’, sectionview.svg has been divided into hatch.svg, which is 2KB; profile.svg, which is 6KB and dimension.svg, which is 13KB. In this representation,for the example shown above, there is a two-way relationship that is maintained between <parent> element and Xlink attributes. Thus, it is possible to navigate the whole hierarchy of the tree on demand. Figure 5 shows the two-way relationship

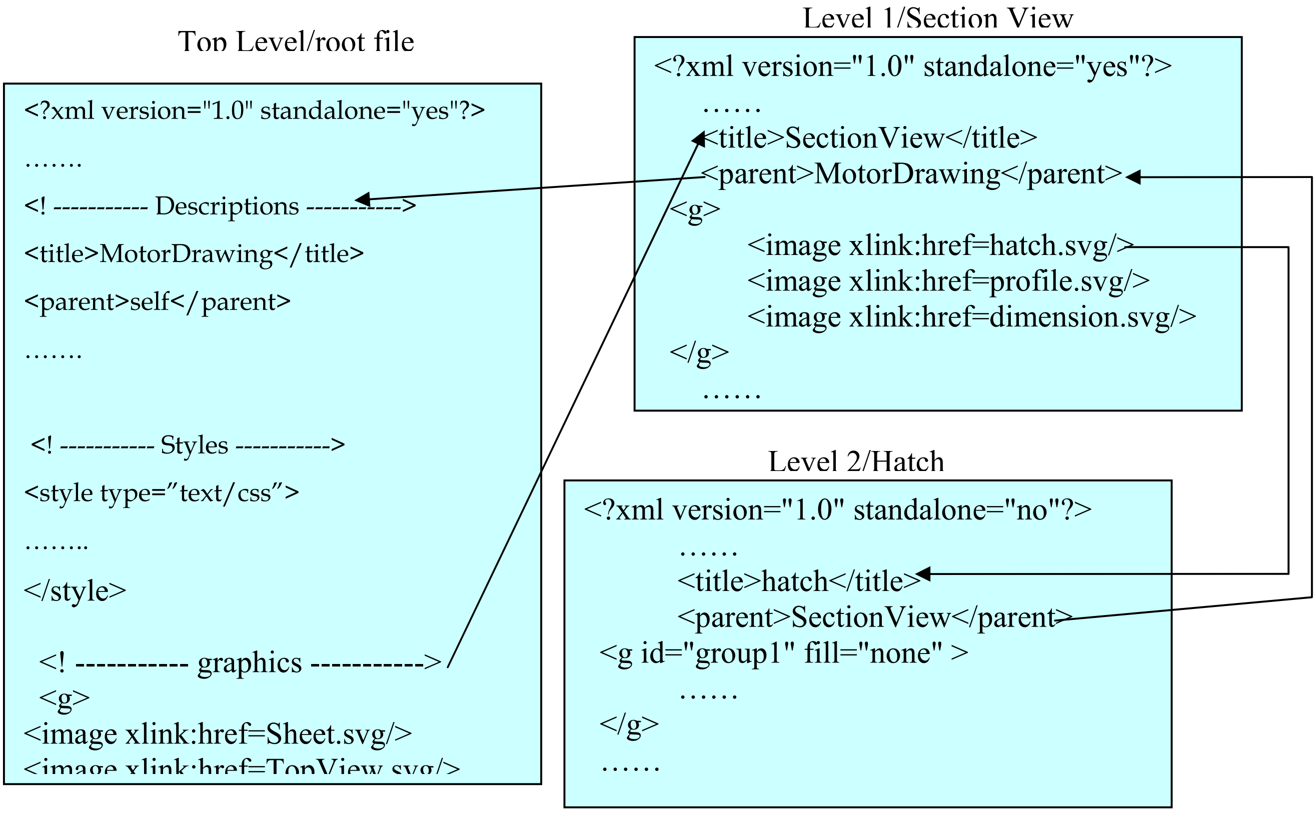


Fig.5. Two-way relationship between content files

5. CONTENT DELIVERY AND INTERACTION

To take advantage of this novel content representation method, we propose an on-demand content delivery and interaction architecture. Two steps of session will be used to represent conversation and interaction between client and content server. First step is called system session and second step is called instance session.

System session starts at the request for any graphic content and begins with the downloading and parsing of the Root of the graphics domain. This information will be stored on the client until the client closes the session and allows the user to navigate to the other levels of the content. Instance session is established when a particular graphics content mapped to links in the root of the graphics domain is requested by the client. This session is maintained only during the file transmission. Session ends when the downloading of graphic content is completed.

Content can be delivered in two ways. The first one is 'sequential delivery'. Once the root is loaded into the memory, system will automatically use the links in the root to discover and download the contents of next level. The second method is 'on demand delivery', where the root is downloaded first, but the system won't download the content of the next level. In turn, the system provides a list of all the content of the next level which can be selected as per the user need. The type of delivery can be chosen by the client as required. During the delivery process, the client's profile will be used to decide which level of content, in what detail and in which format should it be delivered.

5.1 Graphics content delivery framework

Figure 6 shows the schematic of the delivery framework. In this framework, content delivery is initiated by the client request. As mentioned before, by subdividing large content into interconnected multiple small files, content server doesn't need to maintain the state of whole interaction/transaction. Content adaptation and rendering is finished at the client side. The process of delivery is stateless. Advantage of this delivery method is that the server can support more clients at the same time.

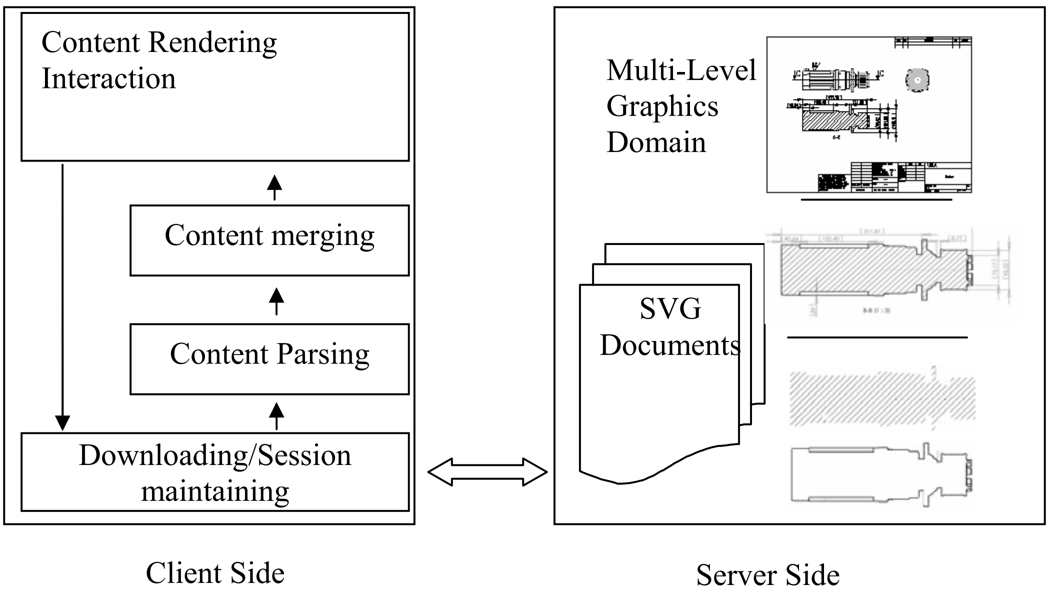


Fig.6. Graphics content delivery and process

Server maintains the graphics domain and listens to client request. Client has four major components: Downloading/Session Maintenance, Content Parsing, Content Merging, and Content Rendering/Interaction. Downloading/Session Maintaining module maintains the conversation and sends content request to server and handles the downloaded content. Content Parsing module will read the content into memory and build a DOM structure for each individual file. Content Merging module merges the different content together. Two of the Content Merging methods will be discussed later. Content Rendering and Interaction module enables content visualization and interaction on the client. It directly controls the Downloading and Session Maintaining module. For example, user can start or end a session, request a portion of content, etc.

5.2 Content merging

Because we use multiple files to represent a graphics domain, each file has its DOM structure built when the file is downloaded to the client. DOM plays a critical role in content merging. Client maintains a DOM structure for each downloaded file during a session. When client renders the content to device screen, we have to merge the content together. In our system it is done in two ways.

The first one is simple content merging. The key idea of this method is to insert the entire sub-level DOM structure into the parent DOM structure and replace the link node which points to the sub-level graphics.

The pseudo code of the method is written as follows.

```
In Parent DOM Structure pDOM
For each elements i
    If elements type is SVG link
        Then find the SVG DOM Structure sDOM
          pDOM.elementi.delete()
          pDOM.AddElement(sDOM)
    end
end
```

The second one is hyper content merging. In this method, DOM structure won't be merged together. When we render the content, the rendering engine will try to load the DOM structure only when it encounters a SVG link node and the user can choose whether to load the SVG DOM or not, based on his point of interest in the overall content.

The pseudo code of this method is written as follows

```
For each elements i
    If elements type is SVG link
        If bDraw is true
          Then find the SVG DOM Structure sDOM
          sDOM.load()
```

```
            sDOM.Draw()
          end
     eles
            pDOM.elementi.draw()
       end
end
```

5.3 Interaction

The mode of interaction is very important for an online collaborative environment. Client will need to add comments, redlines, and other observations to the original graphics content. The user added content will be stored into a separate SVG file and an entry link in the Root file will be created. The Download module will send a request to update the graphics domain on the server so that this change can reflect on the content displayed on the devices of collaborating partners. Then the SVG file (comments, redlines) will be uploaded to level 1 of the graphics domain. The Root file will also be updated. Since we presume the content of Root and Comments file will be very limited, the process won't consume too much bandwidth. This process allows progressive changes to the graphics domain.

6. PROTOTYPE

Based on the framework presented above, a prototype is being implemented, whose overall functionality is shown in Figure 7.

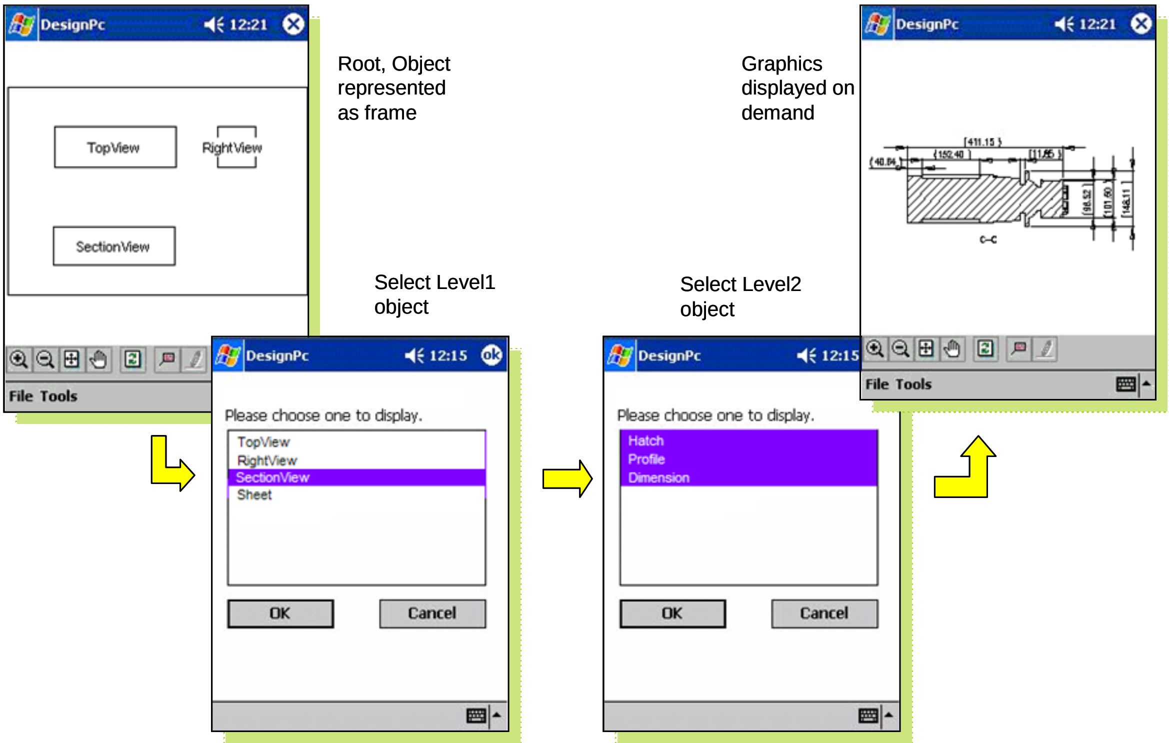


Fig. 7. a prototype of 2D graphics representation

7. CONCLUSION

The endeavor of this research is to investigate novel frameworks and architectures for multimedia content representation and transmission based on industry standards, which would mitigate the problems of bandwidth and mobile computing constraints.

In this attempt we have proposed a SVG based intelligent, distributed, hierarchic and concise 2D graphics content representation and progressive delivery architecture.

The proposed architecture employs a multi-level division method to dissect the graphics content into multiple levels using a variety of rules and strategies. Each level can be delivered on demand as against the delivery of entire monolithic file. Currently, we have implemented the framework. The system has performed satisfactorily in representing different engineering graphics content (drawings, schematics, layouts), which is quite encouraging for field service applications - one of the main focus areas of enterprise application research at WINMEC.

A prototype based on the framework is being developed wherein modules for real-time device based content adaptation, personalized user interactivity, persistent session management and migration, and content synchronization modules will be designed and implemented.

ACKNOWLEDGEMENTS

We are pleased to acknowledge the support of Intel Corporation, and Wireless Internet for the Mobile Enterprise Consortium (WINMEC), UCLA towards partial funding for this research.

REFERENCES

ABELLO, JAMES AND KORN, L. JEFFREY, 2002. MGV: A System for Visualizing Massive Multidigraphs. *IEEE Transactions on Visualization and Computer Graphics 8(1): 21-38*

PRABHU, B. S. AND PANDE, S. S., 1999. Intelligent interpretation of CADD drawings. *Computers & Graphics 23, pages 25-44*

FLORIANI, LEILA DE AND PUPPO, ENRICO, 1995. Hierarchical triangulation for multiresolution surface description, *ACM Transactions on Graphics (TOG), Volume 14 Issue 4.*

JING, JIN, HELAL, ABDELSALAM SUMI, AND ELMAGARMID, AHMED, 1999. Client-server computing in mobile environments, *ACM Computing Surveys (CSUR), Volume 31 Issue 2*

LIM, KAI H., BENBASAT, IZAK, AND WARD, LAWRENCE M., 2000. The Role of Multimedia in Changing First Impression Bias, *Information Systems Research, Vol. 11, No. 2, pp. 115–136.*

JAIN, RAMESH, 2004, Quality of Experience, *IEEE Multimedia, 11, 1, Jan/Mar, pp 95-96.*

LAU, RYNSON W.H., LI, FREDERICK, KUNII, TOSIYASU L., GUO, BAINING, ZHANG, BO, NADIA, MAGNENAT-THALMANN, , KSHIRSAGAR, SUMEDHA, THALMANN, DANIEL, AND GUTIERREZ, MARIO, 2003, Emerging Web Graphics Standards and Technologies, *IEEE Computer Graphics and Applications, Jan/Feb, pp 66-75.*

GUNASEKARAN, A., AND LOVE, P.E.D., 1999, Current and future directions of multimedia technology in business, *International Journal of Information Management* 19, 2, April 1999, pp. 105-120.

GROSKY, WILLIAM I., JAIN, RAMESH, AND MEHROTRA, RAJIV, 1997. The *Handbook of Multimedia Information Management,* Prentice Hall, New Jersey, USA.

ROSENBAUM, R., SCHUMANN, H., AND TOMINSKI, CH., 2004, *Presenting Large Graphical Contents on Mobile Devices – Performance Issues, in: P.C. Deans (ed): E-Commerce and M-Commerce Technologies*, IRM Press, Hershey, London, 2004.

BARAVALLE, ANDRES, GRIBAUDO, MARCO, AND LANFRANCHI, VITAVESKA, 2003, Using SVG and XSLT for graphics representation, In *Proceeding of Open SVG 2003*

CHANG, YI-HONG, AND CHUANG, TYNG-RUEY, 2002. Online Aggregation and Visualization of Census Data: Population Mapping with SVG, XML, and Free Software. In *Proceedings of SVGOpen 2002*, Zurich, Switzerland.

LEWIS, C.T., KARCZ, S., SHARPE, A., AND PARKI, I.A.P.BIOVI, *2002.* Genome Viewer. Development of an SVG GUI for the visualization of genome data. In *Proceedings of SVGOpen 2002*, Zurich, Switzerland.

BARU, CHAITAN, BEHERE, AMIT, AND COWART, CHARLES, 2001, Representation and Display of Geospatial Information: A Comparison of ArcXML and SVG, *Second International Conference on Web Information Systems Engineering (WISE'01) Volume 2*, December 03 - 06, 2001

KOBAYASHI, AREI, TAKAGI, SATORU, AND INOUE, NAOMI, 2003. Extensions of SVG for human navigation by cellular phone, In *Proceedings of the SIGGRAPH 2003 conference on Web graphics: in conjunction with the 30th annual conference on Computer graphics and interactive techniques*

LEIGH, J., JOHNSON, A.E., DEFANTI, T.A., BAILEY, S., AND GROSSMAN, R., 1999. A methodology for supporting collaborative exploratory analysis of massive data sets in tele-immersive environments, *Proc. of The Eighth International Symposium on High Performance Distributed Computing*, Aug 3-6, 1999

FREDERICK, P., GATTASS, M. , AND MEDIANO, M.R, 1999. Efficient visualization of graphical objects, Proc. of XII Brazilian Symposium on Computer Graphics and Image Processing, Oct 17-20. 1999.

SELLIS, TIMOS, ROUSSOPOULOS, NICK, AND FALOUTSOS, CHRISTOS, 1987. The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. In *Proceedings of the 13th VLDB Conference*, pages 507–518, September 1987.

MEDIANO, MAUR´ICIO R., CASANOVA, MARCO A., AND DREUX, MARCELO, 1994. V-tree - a storage method for long vector data. In *Proceedings of the 20th VLDB Conference*, 1994.

GUTTMAN, A., 1984. R-trees: A dynamic index structure for spatial searching, 1984. In *Proceedings of the ACM SIGMOD Conference on Data Engineering*, pages 47–56, 1984.

SU, XIAOYONG, PRABHU, B.S., CHU, CHI-CHENG, AND GADH, RAJIT, 2004, Middleware for Multimedia Mobile Collaborative System, *THIRD ANNUAL WIRELESS TELECOMMUNICATIONS SYMPOSIUM (WTS 2004),* May 14-15, 2004, CalPoly Pomona, Pomona, California, USA

JAIN, RAMESH, 2003, Semantics in Multimedia Systems, *The 9th International Conference on Multi-Media Modeling (MMM '03)*, January 7-10, 2003, Taipei, Taiwan.

PHAM, BINH AND WONG, ON, 2004, Handheld Devices for Applications Using Dynamic Multimedia Data, *Proc. of the 2nd Intl. Conf. on Computer graphics and interactive techniques in Australasia and South East Asia,* June 15-18, 2004, Singapore, pp 123-130.

LEE, SANGMI, FOX, GEOFFREY, KO, SUNGHOON, WANG, MINJUN AND QIU, XIAOHONG, 2002, Ubiquitous Access for Collaborative Information System using SVG, *Proc. of SVG Open / Carto.net Developers Conference*, Zurich, Switzerland - July 15-17, 2002

INTEL, PCA Overview, http://www.intel.com/pca/developernetwork/overview/index.htm

MICROSOFT, Microsoft Mobile, http://www.microsoft.com/windowsmobile/default.mspx

WORLD WIDE WEB CONSORTIUM, http://www.w3.org

OPEN MOBILE ALLIANCE, http://www.openmobilealliance.org/

TI, OMAP for 2G and 3G: Overview, http://focus.ti.com/docs/apps/catalog/overview/overview.jhtml?templateId=1106&path=templatedata/cm/level1/data/wire_omap_ovw

W3C, Scalable Vector Graphics,, http://www.w3.org/Graphics/SVG/

W3C, Extensible Markup Language, http://www.w3.org/XML/

AUTODESK, Drawing Interchange Format, http://www.autodesk.com/techpubs/autocad/dxf/

W3C, About SVG, http://www.w3.org/Graphics/SVG/About.html

LEADTOOLS.COM, Vector Graphics Format, http://www.leadtools.com/Imaging-Developer-Portal/sitemap26.htm

Multimedia File Format Guide, http://www.rice.edu/fondren/erc/howto/formats.html.