

Ubiquitous Access for Collaborative Information System Using SVG

Sangmi Lee¹, Geoffrey Fox², Sunghoon Ko³, Minjun Wang⁴, Xiaohong Qiu⁵

Department of Computer Science
Florida State University, USA
slee@csit.fsu.edu¹

Department of Computer Science
Indiana University, USA
gcf@indiana.edu²

Pervasive Technology Laboratory
Bloomington, IN, USA
shko@indiana.edu³

Department of Electrical Engineering and Computer Science
Syracuse University, USA
mwang03@csit.fsu.edu⁴,
xiqiu@syr.edu⁵

Tel: 812-856-1324

Postal address: Indiana Univ. Research Part,
501 North Morton Street, Suite 222
Bloomington, IN 47403, USA

Abstracts

With advances in wireless networking and Moore's law driven advances in computing capabilities, collaborative systems are needed to explore the large volume of distributed resources and new styles of communication. To achieve the goal of collaborative service for users with heterogeneous devices, the system needs to consider not only the traditional design issues, but also the heterogeneity of user devices. We describe the system architecture of our Garnet collaborative system based on our event brokering system, Narada. We discuss our approaches to shared resources with W3C SVG along with initial performance results.

Keywords: Collaborative system, SVG, Event Service, resource sharing

1. Introduction

There are growing opportunities for collaborative use of hand-held and desktop devices. In distance-education students can use Personal Digital Assistant (PDA) with real-time collaborative digests of a lesson to make notes. In research, mixing of PDA and conventional interfaces to computers provides new research environments. We have designed and built a collaborative information system, *Garnet* [1], which has several collaborative tools including the functionality of shared graphical information represented by Scalable Vector Graphics (SVG) [12]. This is an example of what we call shared export – we choose important format of representations supportable several applications. We then build a custom shared event browser for this representation. The shared event model for collaboration is particularly important for sharing with heterogeneous clients as it allows different customizations for different clients. The more common shared display model only offers limited customization. SVG is natural for all 2D vector

graphics as in dynamic presentation (curricula pages), scientific visualization and whiteboards. Collaborative SVG is important as it illustrates the possibilities of sharing documents supporting the W3C DOM with its powerful and general event structure.

2. Related Work

Web based collaboration systems support universal accessibility to the devices with available Web browsers. There are various Web-based collaboration systems including commercial products such as WebEx Communication's *WebEx* [5], Centra Software's *Centra* [6], and PlaceWare's *PlaceWare* [7]. However, small wireless devices are limited in their computing capabilities and their Web browser is not fully functional for current advanced Web technology. University of Pittsburgh's *Pebbles PDA project* [28] is suggesting a model of collaboration between heterogeneous PDAs.

Several collaboration systems enhance their accessibility with XML and XML based markup languages. Jabber. Inc.'s Jabber [21] is an instant messaging and presence-enabled applications and service with XML-based communication. Annotea [17] from W3C is a Web-based shared annotation system including SVG annotation. Annotea describes and manipulates the annotation with RDF schema. The universal W3C DOM document representation of information provides a key to interoperability in a heterogeneous environment. Annotea allows general collaborative viewers to support collaborative annotation for desktop computers, which is anchoring to objects within the SVG document. Meanwhile, Garnet system supports collaborative annotation at general pixel positions, which are then rendered at the appropriate scaled position on each client synchronously.

3. Garnet System Architecture and Integration for Garnet and Hand-held device

3.1 System Overview

We have been investigating strategies to build web-based support for people to interact with each other and with other resources including hardware, documents, and applications. This was originally called a *Collaboratory* by Bill Wulf in a famous Science article in volume 261, 13 Aug 1993 [29]. We must do this while technology is rapidly changing and while we are not certain what collaborative tools. Scientists and students will actually use i.e. the requirements are not known. We have been focused on a set of successful capabilities where some consensus exists as to what they do and how they look to users – these are typically commercialized. Meanwhile, support of hand-held devices is so poorly understood that in spite of strong commercial interest, it remains a good research area.

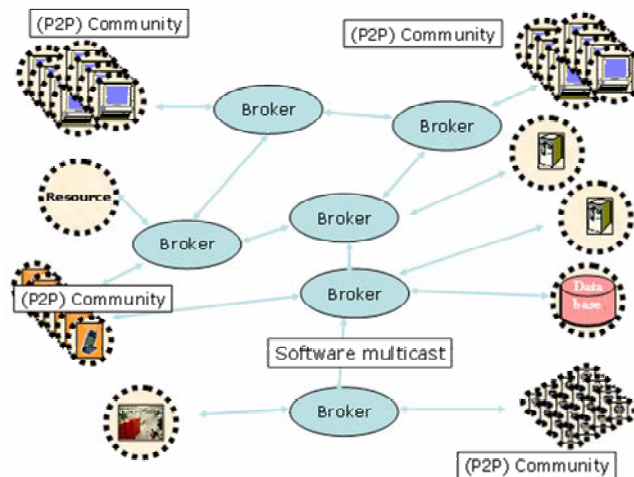


Figure 1. Distributed Brokers Implementing Event Service

We suggest that there are now some pretty good distance education and more general collaboration solutions and that now is a reasonable time for groups to invest in learning and using some of the tools. Clearly capability, performance and robustness will improve but there seems to be consensus in several areas. Garnet system is designed to support asynchronous and synchronous collaboration [1]. Garnet's features of the web and distributed objects are the natural support of asynchronous collaboration. One can post a web-page or host a Web Service and then others can access it at their convenience. Search and registration capabilities are a good asynchronous environment. The field of synchronous collaboration is very active at the moment and we can identify several important areas:

- 1) Basic Interactive tools including Text Chat, Instant Messenger, and White boards
- 2) Shared resources including shared documents as well shared visualization
- 3) Audio-video conferencing

Those features are implemented based on a distributed publish-subscribe model for coordinating the collaboration with a distributed event brokering system. Figure 1 depicts the structure of the distributed brokers. Every broker supports the interaction between users using a message object model. Individual brokers maintain network snapshots and make intelligent decisions aiding the efficient disseminations within a vast broker network. This uniform event bus is capable of supporting both centralized and peer-to-peer collaboration models. This event bus supports both control messages and events specifying changes in object state. This is the conventional shared event model of collaboration where such object state messages ensure that objects are properly shared between collaborating users. Each state of objects can be shared by the users with various methods. We will describe this in detail in section 3.3 in detail. We developed a distributed event brokering system Narada [22] that can deliver events either in a deterministic reliable way like JMS or in the more opportunistic JXTA peer-to-peer [26] model. Narada also interpolates smoothly between these models and would normally be run in a fashion looking like a distributed dynamic set of JMS style event brokers. However all results in this paper come from work using the commercial SonicMQ JMS event broker. The resultant event service is called the GMS whose XML specification is named with GXOS [23].

3.2 Integrating Hand-held devices to Garnet

One of the interesting design issues of the Garnet system is the integration of heterogeneous devices. PDAs, intelligent mobile phones, as well as conventional desktops. We classify and define the heterogeneity among various devices

- Network environments: Stable and reliable network service is needed to support the internetworking in heterogeneous wireless technologies and conventional wired LAN environment
- Computer hardware: Variation in CPU processing powers. The size of memory and constraints imposed by the display system should be taken into account especially for the light weight devices such as PDAs or intelligent phone.
- Operating Systems: Just as differences in computer hardware cause heterogeneity of data representation, different operating systems cause different representations. Moreover, some of light-weight Operating Systems support a very constrained act of particular network communication protocols only.

Web-based systems allowing the users to access their resources via browsers, also meets the users' need of universal accessibility. There has been an early approach to the goal of universal accessibility from W3C, the Web Accessibility Initiative (WAI), which develops guidelines for accessibility of Web sites, browsers and authoring tools. W3C and Oasis' Web Service for Interactive Applications (WSIA) defines the Web Services interfaces and XML vocabularies to deliver web applications to end users through a diversity of deployment channels: directly to a browser or mobile device, indirectly through a portal, or by embedding into a third party web application. Moreover, it is aiming to efficiently create web applications that can be easily be modified, adapted, aggregated, coordinated, synchronized or integrated.

Similar to other distributed object systems, Web Services define an interface definition language: Web Service Definition Language (WSDL). Web Service also defines Universal Description, Discovery, and Integration (UDDI) [24] or Web Services Inspection Language (WSIL) [25] as naming and discovering system for distributed objects. Figure 2 illustrates a schematic representation of the workflow of an application invoked through portlet.

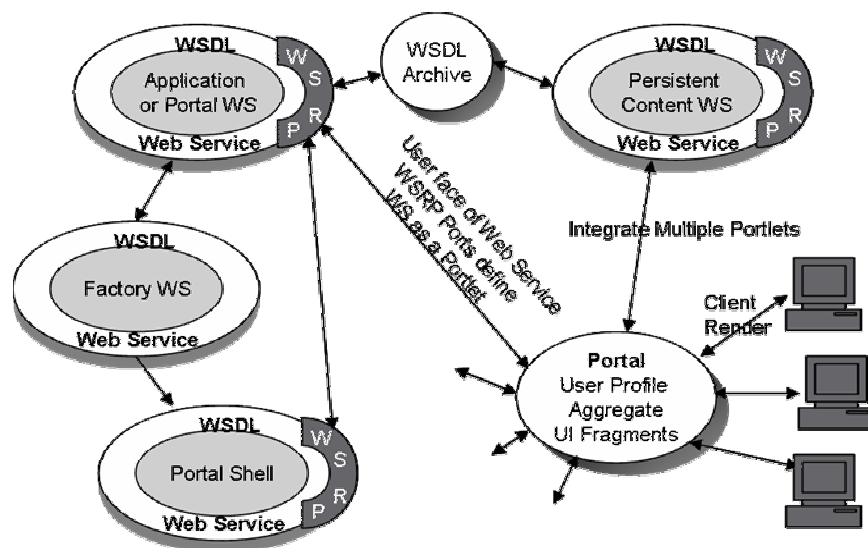


Figure 2. A schematic representation of the lifecycle of an application invoked through a portlet

This approach provides a separation between the server that manages the user interface and the server that manages a particular service. This separation is not present in the three-tiered portal model and is the major development for breaking the event processing pipeline for Web-based collaboration. Currently we are investigating the different Garnet capabilities as different portlets with SVG shared export as a collaborative Web Service. However, we will not discuss this here.

To integrate various devices in the presence of aforementioned heterogeneity, we propose a new middleware supporting universal accessibility to the Garnet System; Garnet Message Service Micro Edition (GMSME) [26]. This is designed to support internetworking between wireless network and wired LANs, and the less powerful CPU and memory size in such devices. Garnet system exploits pure Java solution to inherit the universal accessibility features already built in to the Java virtual machine.

The major components of GMSME are a PDA adaptor, a Hand Held Message Service (HHMS) protocol, and an API for application processors, which does message processing on the mobile device application, such as shared SVG export [12, 15] or shared display described later. The PDA adaptor sits between mobile clients and the GMS system and does user management, message mapping, connection management, and message optimization based on the user and device specification. The PDA adaptor ‘listens’ to the messages and events on the Garnet collaboration system by subscribing to GMS and it delivers messages from the mobile clients to GMS by publishing regular GMS messages that it creates from messages received from mobile clients.

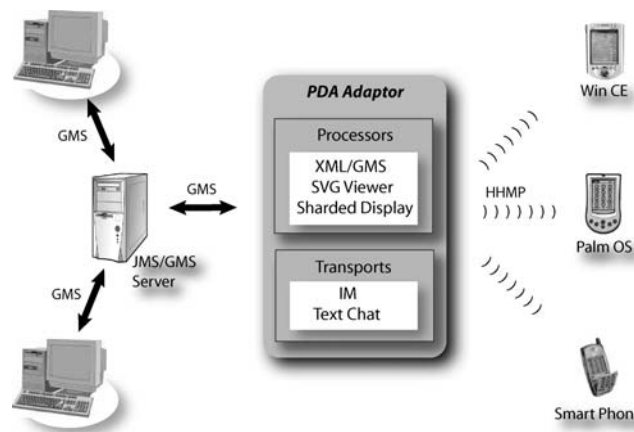


Figure 3. The structure of PDA Adaptor of GMSME

GMSME provides an optimized service, HHMS, considering the constraints of hand held devices. The PDA adaptor parses the GMS message body which is described in XML syntax and delivers it to the mobile client over HHMS. As we mentioned in the beginning of this section, mobile clients have limited performance and size in comparison with desktop computers. Therefore, they require specially tuned protocols. The HHMS protocol is designed for light weight but effective communications. The structure of HHMS is shown in Figure 3. The general GMS services are mapped with HHMS services. HHMS also sends and receives the messages containing event and data between software applications. Table 1 provides a comparison of GMS and HHMS in terms of their services.

GMS over JMS	HHMS of GMSME
<i>Publish message</i>	SendHHMS and SendGMS. Mapping the messages

	between GMS system and mobile clients.
<i>Subscribe message</i>	Register and login
<i>MessageQueue</i>	JobQueue keeps the load inside limits
<i>Recovery</i>	Logfile records major events

Table 1. Comparing message services in GMS and HHMS

GMSME supports heterogeneous clients by customizing GMS messages depending on the client specifications. We define and store user profile and mobile device profile, such as style sheets designed for particular device. As with other objects in the Garnet collaboration system, user profile and device profile use the XML object metadata specification GXOS [23]. As a result, we seamlessly exchange information between the conventional Garnet collaboration systems and the mobile device. In addition, these profiles make it possible to render messages differently for the mobile users. For example, the shared export update of SVG, PDF, or HTML to a non-color and smaller-size display mobile device can be rescaled and have color modifications to provide personalized and optimized look.

Considering the performance issues for GMSME, the performance of a mobile device is much more limited than that of a desktop's. We accomplish optimizations by two design strategies. First, we move processes to server side. To reduce process load of mobile device, we move the processing module for each application from the hand held device to the PDA adaptor on the server side. For example, before sending an event message, such as a shared export update, the PDA adaptor performs graphic processing needed for SVG shared export events. As a result of this processing, the mobile client receives a ready to use image from the PDA adaptor. Second, we use the optimized protocol, HHMS. HHMS is a byte oriented message protocol between a mobile client and the PDA adaptor on desktop. Tag bytes in the byte array encode application types, event types, and event message.

3.3 Sharing Resources in Collaborative System

Sharing resources is an important service to support collaborating users. Shared resources includes any kind of shared objects such as shared documents, shared data streaming, and shared visualization. The objects can be shared in several ways. We discuss three object-sharing styles;

- Shared display: The system shares the bitmap defining display window of the collaborative application.
- Native Shared Object: The object is accessed directly and its state is changed by collaborative applications. The collaborating participants receive the message defining the changed state, and maintain consistency between the shared object's representations on the different machines.
- Shared export: The object is converted to some standard form that is more flexible than bitmap of shared display. This form makes it possible to be used by several applications.

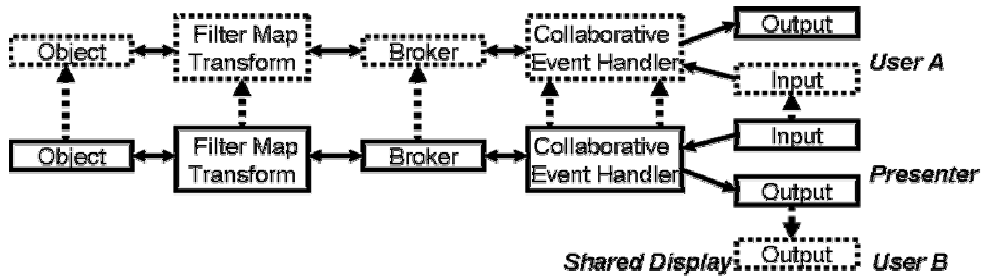


Figure 4. The visualization pipeline for shared object

As an example, Figure 4 illustrates a visualization pipeline with three users; presenter, user A and user B. Presenter shares the object with users A and B. The output and input wend their way through multiple filters (tiers) until they are finally rendered on the particular client device which could be different for each user. As the simplest case, user B is sharing presenter's object with shared display. User B shares the final frame buffer, which is exactly the same bitmap as the presenter's view. As shown by vertical arrows, the user A accesses objects in format separated with presenter's bitmap view. For instance, User A does not access to the pixel data of presenter. User A accesses the object already filtered to the document which describe presentation without its pixel data. Therefore, user A has maximal flexibility as he or she can choose to use or ignore presenter's visualization state change. In particular user A does not have to use the same display device as presenter; B could be high end CAVE, User A on PDA. Shared export corresponds to one of the intermediate arrows where one is inside the pipeline at a stage where the format is some standard such as HTML, PDF, Java2D, or 3D, SVG. Then one can build a generally useable collaborative viewer for this intermediate form and produce a powerful environment in a re-usable fashion.

Garnet System supports the shared display as a method of sharing application. It is the simplest method as seen on Figure 3. Moreover, GMSME modifies the output to fit for lower resolution display systems. Not only for the customizing resolution, has it optimized the data compression methods and incremental updating styles. However, shared display still does not allow significant flexibility. Wireless devices need a wide variety of viewpoints due to their mobility and limited display systems, and separating the presentation from data object is necessary to fulfill the needs of various viewpoints. More flexible sharing is possible by sharing object state updates among clients with each client being able to choose how to realize the update on its version of the object [1]. This is very time consuming to develop if one must do this separately for each shared application. The shared export model filters the output of each application to one of a set of common formats and builds a custom shared event viewer for these formats. This allows a single collaborative viewer to be leveraged among several different applications.

The format which defines the object is one of important design issues in this phase. The scalable formats W3C's Scalable Vector Graphics (SVG) [12] and Adobe Inc.'s Portable Document Format (PDF) [13] are particularly interesting and support of collaborative viewers for them is a major advantage of Garnet System. Scalability implies that each client can resize and scroll while preserving correct positions of pointers and annotations for their various resolutions. SVG is useful as it is already available for Adobe Illustrator [14] and both PowerPoint and Macromedia Flash are exportable to this syntax. Currently there is a Flash (which is a binary 2D vector graphics format) to SVG converter [15] from the University of Nottingham; OpenOffice.org's *OpenOffice* [16] exports PowerPoint to SVG.

We would recommend building SVG exports into tools like whiteboards and 2D scientific visualizations to allow convenient interchange among these different 2D presentation tools [1]. We describe the design issues of shared export with SVG format, structure of the event processing, and implementation in Garnet system in following sections.

4 SVG Shared Export

4.1 Authoring Presentation in SVG

We have discussed how hand-held devices are integrated to the Garnet distributed distance education system. In this section, we will explain how we adapt SVG as a format of shared export. The first stage of the Distance Learning Service will be preparing the slides for the presentation. The object of the shared presentation object is authored in various ways with various formats. The users can create their presentation with popular commercial authoring tools such as Word Processor, PowerPoint, Webpage editor or some multimedia page editor such as Macromedia or Adobe tools. For the purpose of education and training, these current tools are not well tuned. The authoring for education usually needs to change the content rapidly, create lots of similar pages, and generate a presentation tools supporting some interactive functionality. As the Web authoring is improved with the powerful XML based systems using XSLT style sheets, we expect XML based information format will provide convenient authoring style and interoperability among current authoring tools.

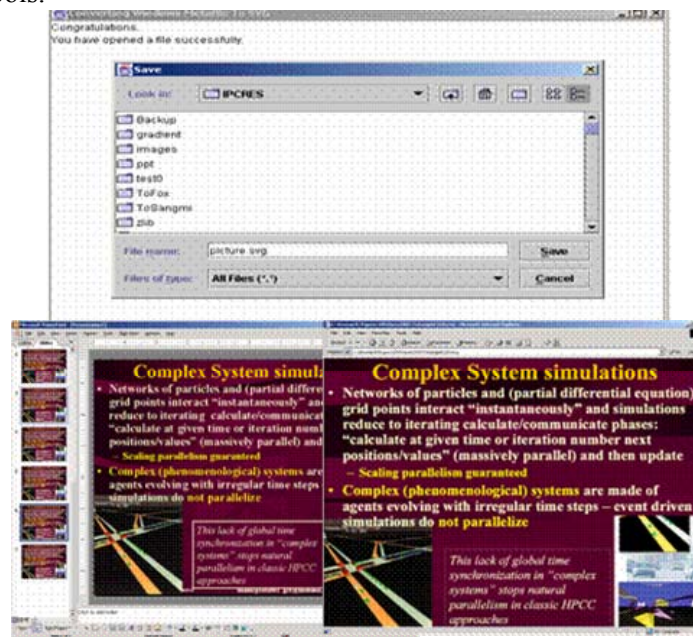


Figure 5. Screen Capture of Converting Tool

We developed tools which translate HTML document file including Vector Markup Language (VML) [18] tags, and WMF binary file to SVG document file. The users can create their presentation as a SVG document from the presentation slides generated by MS Office. As an initial effort, we used HTML document files which describe the graphical information with VML, a precursor of SVG, exportable by MS Office. We developed the application converting VML elements to SVG elements. However, we realized that this conversion is less elegant and efficient. Currently, we provide the tool of translating through Windows Metafiles (WMF) to generate SVG document file. Since MS Office is offering the feature of exporting to WMF, the users can export their file to WMF and convert the file to SVG after they finish the authoring with MS Office applications.

This converting tool supports all types of SVG document information: image, graphics, and text. The image included in PowerPoint slide is translated to SVG image element with its original format, and the graphics defined in WMF specification are translated to SVG graphical elements. The types of font provided in PowerPoint are fully supported in this tool. Figure 5 shows a screen

capture of the converting tool. If the user types the name of input, and output files, this tool automatically translate files.

4.2 Shared SVG Browser

One of the goals of Garnet system is shared resources. In Garnet system, the objects of resource are shared in two ways: shared display, and shared export. As mentioned in the previous section, shared export allows more significant flexibility to collaboration service users. The Garnet system provides an integrated SVG browser supporting flexible sharing and collaborative event sharing for desktop computers and small wireless devices. The requirements for the shared SVG browser for Garnet system are listed (given in no particular order):

- Narada/GMSME Message types. The event message processed in shared SVG browser should be a Narada message [22]. For the small wireless device version, each personal server delivers their message to PDA clients as GMSME message type. However, the personal server, as a conventional desktop client should process Narada message to communicate with other desktop clients.
- Support various network protocols. Since the devices are of a wide variety, the network communication method cannot be unified. The shared SVG browser should be able to share event and resources in this heterogeneous network environment.
- Customized presentation depends on device style. Shared SVG engine in Personal Server produces customized display with stylesheets designed for specific device display style.
- Annotated documents are well-formed, structured documents. The annotation event appends a set of elements to the original document.
- Annotations must be described in SVG graphics. Most of annotation systems depict their pointer or annotation as an image. Using vector graphics for presenting shared pointer or annotation provides flexible presentation to users with extremely small display device such as a cell phone or a palm watch.
- Collaborative event handling. Garnet Shared SVG browser processes collaborative events synchronously. Garnet should make it possible for any device to connect to the Garnet system.
- Two phased coordinate system in collaborative event. There are two coordinate systems used in Shared SVG Browser, original SVG document coordinate system and bitmap based coordinate system. User can interact with the coordinate system regarding their bitmap based display, and the SVG collaborative engine on the Personal Server translates it to the original SVG coordinate system. The value of the (x, y) position processed with different devices is translated to the value regarding the original SVG coordinate system to produce the consistency.
- Improve the performance. Each image produced by the Personal Server is compressed and delivered to PDAs.
- Reliable application protocol. The messages between desktop computers are using the service of Narada based on MOM with message queue and publish/subscribe method. It provides reliability to deliver message. However, the optimized protocol between Personal Server and PDAs does not guarantee their delivery. Thus, the application should consider the protocol for reliable message communication.
- Fully integrated to Garnet system. This shared SVG browser is integrated as a feature of Garnet system.

In Garnet system, the shared SVG browser is invoked when the presenter loads SVG document to his or her browser. The SVG collaborative engine sitting on presenter side sends an initial event message. The message includes initial event and initial URL in string. The initial message from the Narada event brokering system is published by GMS service. Finally, the SVG collaborative

engine on the client side processes the initial event, opens the shared browser with the URL that presenter sent, and renders the image from the downloaded SVG document.

The Shared SVG browser for desktop computer has the following components.

- Collaborative Event Handler
- DOM tree Generator
- SVG Image Renderer
- Collaborative Browser

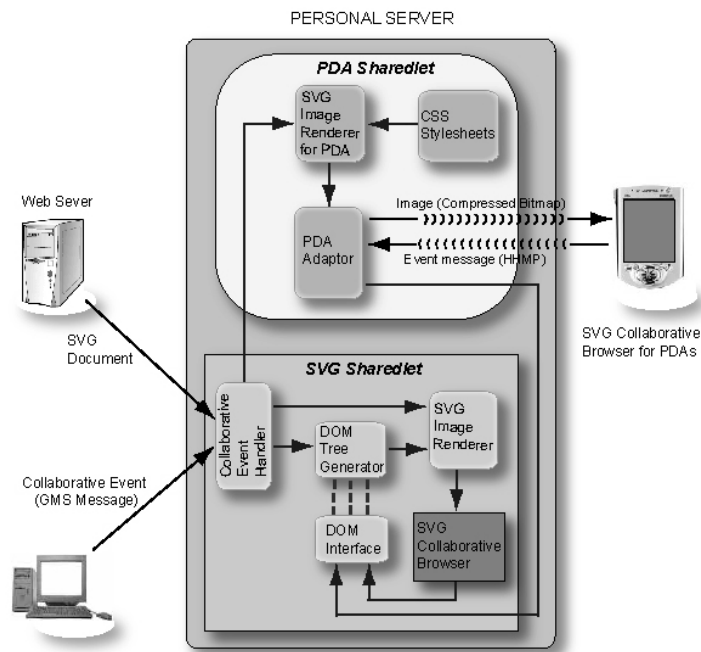


Figure 6. The workflow of the Shared SVG browser in Garnet system

The Collaborative Event Handler parses event messages received from other participants. As is depicted in Figure 6, Event Handler accesses Web Server and downloads the SVG document to local memory. If the presenter loads a new document and if it is the first time that the presenter publishes, the Collaborative Event Handler invokes the client's collaborative browser automatically.

The DOM tree Generator generates initial DOM tree from downloaded SVG document. We used Apache's APIs to parse SVG documents and generate DOM tree. This DOM tree is accessible from the PDA Adaptor of the Personal Server. The image from SVG document is rendered by SVG Image Renderer. The SVG Image Renderer generates the image dynamically, for instance,

if the document is changed, it renders only the changed parts. Finally, Collaborative Browser shows the images rendered by SVG Image Renderer. Moreover, it is a user interface realizing the input events.

The Personal Server is a conventional Garnet client too. Thus, Personal Server includes shared SVG browser units already mentioned. Additionally, it has CSS2 [8] stylesheets for various small wireless devices, and a PDA event handling unit. It shares downloaded SVG documents and DOM tree with the shared SVG browser unit for conventional desktop applications. However, it renders its image with CSS2 designed for small wireless devices. The customized image is passed to PDA event handler unit. Note that there is a network connection between PDAs and Personal Server since the PDAs have connected to the Personal Server.

The PDAs gets only ready to use image from Personal Server. We have investigated to find appropriate graphic format to get better performance and quality image. Currently, we use raw bitmap data to illustrate the image and compress it with Huffman and LZ77 encoding to improve the performance. It is the same way to compress image data with PNG [4] image format. Java J2ME CDC [3] is supporting PNG image format which is the lossless graphic format for network communication. Although J2ME CLDC [10] is designed for more powerful devices, Java J2ME CLDC does not support PNG format.

In the next section, we will describe the collaborative event handling in detail.

4.3 Collaborative SVG Event Processing

As already mentioned, Garnet system supports shared resources i.e. shared objects, which can be documents, applications, data streams or visualizations. The object can be shared in several ways, which trade off ease of use versus flexibility versus ease of implementation. The shared SVG browser is our prototype of shared export to get more flexibility than bitmap of shared display. Compare to the shared display which processes the events containing updates to frame buffer, shared export processes various event types. The Garnet System provides the interactive framework for collaborative SVG export. In the shared SVG browser, we can classify and define main collaborative events,

- Application Invocation: The most initial event of shared SVG browser.
- Resolution Customization: If the presenter zooms in or out of their SVG image, other participants' shared SVG browser will render graphics based on the resolution from the presenter.
- Annotation: The presenter's annotation will be shared in participants' browser. The event annotation will contain the X, Y 2D graphical positions on the current bitmap and annotation type.

The application invocation event is processed when the presenter opens their shared browser and loads new SVG documents from new URI. This event message includes the type of event and new URI for SVG documents. Other participants joining the session will get the event message and access the URI, and load new document. Each shared SVG browser parses the document for its own presentation. Every participant maintains its own replica of SVG document and DOM tree.

The presenter can show zoom in or out image to other participants. The resolution customizing event message includes the rate of zooming and (x, y) position that the presenter pointed as a center of image. Every value of position is transformed with SVG document coordinate system. The pointing event from PDAs is processed in Java canvas user coordinate system. For instance, the user cannot realize the SVG document coordinate system. The user points with own presentation and coordinate system. Meanwhile, the event message between participants only

includes SVG coordinate system based values. This coordinate system transform is processed for all events in SVG shared browser, such as annotation. Each participant will generate customized presentation with the values in event message and CSS stylesheets designed for specific display system.

Annotation is one of basic functionalities for collaborative shared export. The user can choose the type of annotation. After choosing the type, the user points to the position that will be annotated. Annotation event message contains the type of annotation and values of the position. Each participant has the annotation elements group encoded by SVG syntax. Therefore, the annotation pointer in Garnet shared SVG browser adapts the advantage of SVG graphics. For the small wireless devices, the annotation is able to be customized with the stylesheets also i.e. the size or transparency of point. The participants can edit their annotation pointer with any SVG editor. After receiving the event message including x, y position and annotation type, the participants generate SVG document and add the elements to their own DOM tree.

4.4 Performance Evaluation

At this early stage of our project, we have decided to restrict performance measurements to those that will give some indication of the basic communications performance. In order to examine preliminary analysis of *asynchronous* GMS and *synchronous* GMSME communication frameworks, we evaluate the average throughput for both GMS server and PDA adaptor. We used publish/subscribe with non-persistent/non-durable mode for GMS and the one-to-one for GMSME. A non-persistent delivery and a non-durable subscription are JMS modes that do not preserve message in a case of any system failure.

The system environment for benchmarking two communication frameworks is as follow:

- GMS :
 - JMS Server : Sun Ultra60 Workstation, Solaris 8, Dual 450Mhz UltraSparc CPUs
 - Garnet Clients : Windows 2000, 512 MB RAM
 - Networking : 100BaseT Ethernet
 - JMS : SonicMQ E-Business Edition 3.5
 - Java : Sun JDK1.3.0 for Server and Sun JDK 1.3.1 for Clients
- GMSME :
 - Adaptor Server : Windows 2000, 512 MB RAM
 - PDA Client : iPaq H3650, WinCE 3.0
 - Networking : Wireless 11 Mbit/s IEEE 802.11b compliant PCI
 - Java : Sun JDK1.3.1 for Adaptor and PersonalJava1.1 for iPaq

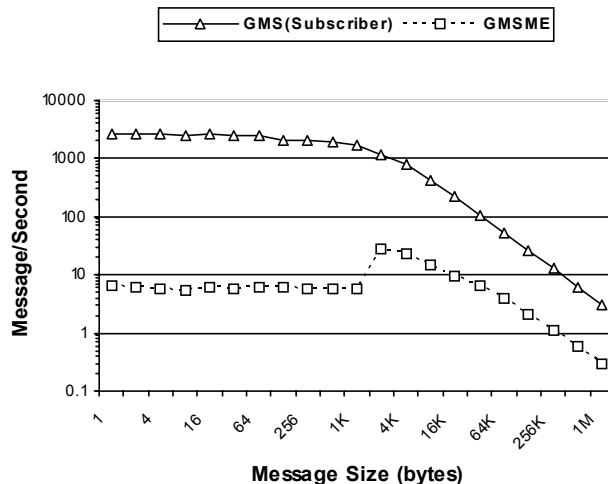


Figure 7. Comparison of GMS and GMSME for real-time collaboration for events defining collaboration synchronization

Figure 7 shows the average message throughput per second for different message sizes. It is clear that GMSME is significantly slower than SMS messages. This is due to low bandwidth of wireless network, synchronization overhead and the slower processing of PDA. In our benchmark, GMSME reaches maximum throughput when message size is 4KB. This indicates that we would expect best performance for PDA Shared Display when an image is divided into 4KB pixel block sizes. An image block optimization was adopted in Garnet Collaboration Shared Display and will be applied to PDA system. We are also investigating more efficient GMSME implementation by using queue mechanism and a new NIO [27] API that provides a multiplexed, non-blocking I/O facility.

5 Future Work and Conclusions

Garnet system provides integrated SVG browser supporting flexible resource sharing and collaborative event sharing for desktop computers and small wireless devices. Compared to the shared display Garnet also featured, SVG shared export provides better quality graphics to users and document based interactivity to developers. In spite of higher complexity in the initial development, we expect that the separation of the presentation and the original document will offer the more flexible viewpoints to users of scientific visualization. Moreover, the DOM based processing is expected to provide a great possibility of scaling systems with its simplicity and interoperability.

Currently, we are investigating to extend to other mobile devices besides our initial effort, Compaq's Ipaq [9]. We extend this work to the devices with more limited resources, defined CLDC by J2ME, such as cellular phones or PalmOS [11] based PDAs. Thus Short Message Service (SMS) can be added for these mobile devices for sending brief messages. The fault tolerance approach is one of critical issues for this system due to the unstable wireless network. We introduce job queue to the PDA adaptor. The job queue will keep the load on mobile device below system limits. This allows us to queue messages that otherwise would exceed available bandwidth and also to send high priority messages by changing message transmission order. Of course, we need a fault tolerant approach for PDA adaptor itself. As we extend Garnet to be accessible from more devices, developing customized stylesheets for each device is needed. CSS defined stylesheets for small and different types of display system, nevertheless, they can not support all of the various devices with new types of display systems currently produced. Moreover, the prototypes of SVG shared export discussed in this paper can be extended to collaboration of dynamic SVG.

The use of Web Services is expanding rapidly as the need for communication and interoperability between the applications is increased. The Web Service supports separation between the server that manages the user interface and the server that manages a particular service. This concept is expected to provide more interoperability in our collaborative system. When we define the SVG shared export as a portlet of the Web service, we should look at the visualization pipelining (section 3.3) more closely. An important design issue is that we must address the break point of the event pipeline to provide interactivity to the SVG component. The computing capability and network bandwidth will be the major factors to decide the breaking point on the pipeline. Moreover, each document in every event states should be clearly defined with appropriate types of DOM and the interface.

References

- [1] G. C. Fox. "Architecture and Implementation of a Collaborative Computing and Education Portal". ERDC Technical report, May 2001 <http://aspen.ucs.indiana.edu/collabtools/updatejuly01/erdcgarnet.pdf>
- [2] G. C. Fox. "Ubiquitous Access for Computer Science and Education". ERDC Technical report, May 2001, <http://aspen.ucs.indiana.edu/collabtools/updatejuly01/erdepdta.pdf>
- [3] Sun Microsystems. Connected Device Configuration: Java 2 Platform Micro Edition. Sun Microsystems, version 1.0 edition, May 2000. Available at <http://www.sun.com/software/communitysource/j2me/cdc/>
- [4] M. Adler et al., PNG (Portable Network Graphics) Specification, W3C Recommendation, October, 1996, Accessed from <http://www.w3.org/TR/PNG-Credits.html>
- [5] WebEx Communication, Inc. WebEx. <http://www.webex.com>
- [6] Centra Software, Inc. Centra. <http://www.centra.com>
- [7] PlaceWare, Inc. PlaceWare. <http://www.placeware.com>
- [8] B. Bos, H. W. Lie, C. Lilley, Jacobs, I., ed. Cascading Style Sheets, level2 CSS2 Specification, W3C Recommendation, May, 1998. Accessed from <http://www.w3.org/TR/REC-CSS2/>
- [9] Compaq, Inc. Compaq iPAQ Pocket PC 3765 Detailed Specifications, <http://athome.compaq.com/showroom/static/iPaq/3765.asp>
- [10] Sun Microsystems. Connected, Limited Device Configuration: Java 2 Platform Micro Edition. Sun Microsystems, version 1.0 edition, May 2000. Available at <http://java.sun.com/products/cldc/>.
- [11] Palm, Inc. Palm m505 Handheld Product Details, <http://www.palm.com/products/palmm505/details.html>
- [12] J. Ferraiolo, ed., Scalable Vector Graphics (SVG) 1.0 Specification. August 2000. <http://www.w3.org/TR/2000/CR-SVG-20000802/>.
- [13] Adobe Systems, Inc. Portable Document Format. <http://www.adobe.com/products/acrobat/adobepdf.html>
- [14] Adobe Systems, Inc. Adobe Illustrator. <http://www.adobe.com/products/illustrator/main.html>
- [15] SVG and the Electronic Research Group, Flash to SVG, <http://broadway.es.nott.ac.uk/projects/SVG/#project>
- [16] OpenOffice.org, OpenOffice. <http://www.openoffice.org>
- [17] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. "Annotea: An Open RDF Infrastructure for Shared Web Annotations." In Proc. of the WWW10 International Conference. Hong Kong, 2001.
- [18] B. Mathews et al., Vector Markup Language (VML), WWW Consortium, May, 1998. Accessed from <http://www.w3.org/TR/NOTE-VML>
- [19] Palm, Inc. Palm m500 Handheld Product Details, <http://www.palm.com/products/palmm500/details.html>
- [20] E. Christerson, F. Curbera, G. Meredith, Weerawarana, S., Web Services Description Language (WSDL) 1.1, W3C Note, March, 2001. Accessed from <http://www.w3.org/TR/wsdl>
- [21] P. Saint-Andre, Jabber Server v1.2 Technical White Paper, Jabber.com, Inc., http://www.jabber.com/pdf/Jabber_Server_White_Paper.pdf
- [22] G.C.Fox, and Pallickara, S., "The Narada Event Brokering system: Overview and Extensions", In Proc. of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02). <http://grids.ucs.indiana.edu/users/shrideep/narada/>
- [23] PTLIU Labs, GXOS <http://aspen.ucs.indiana.edu/~balsoy/gxos/>
- [24] UDDI org, Universal Description, Discovery and Integration, Accessed from <http://www.uddi.org/specification.html>
- [25] P. Brittenham, An overview of the Web Services Inspection Language, Accessed from <http://www-106.ibm.com/developerworks/webservices/library/ws-wsilo/>
- [26] G. C. Fox, S. Ko, K. Kim, S. Oh, and Lee, S., "Integration of Hand-Held Devices into Collaborative Environments", to appear in the proceedings of the 2002 International Conference on Internet Computing (IC-02). June 24-27 Las Vegas., Accessed from <http://grids.ucs.indiana.edu/ptliupages/publications/pdagarnetv1.pdf>
- [27] Sun Microsystems Inc., NIO APIs <http://java.sun.com/j2se/1.4/docs/guide/nio/>
- [28] Brad A. Myers, "Using Hand-Held Devices and PCs Together.", Communications of the ACM. Vol. 44, Issue 11. Novembaer, 2001. pp. 34 - 41
- [29] Wulf. W., "The Collaboratory Opportunity", Science, v 261, 13 Aug 1993

