Integration of Hand-Held Devices into Collaborative Environments

Geoffrey Fox, Sung-Hoon Ko
Community Grids Laboratory, Indiana University
224 Showers Bldg.
501 N. Morton St.
Bloomington, IN, U.S.A.

Kangseok Kim, Sangyoon Oh
Computer Science Department, Indiana University
222 Showers Bldg.
501 N. Morton St.
Bloomington, IN, U.S.A.

Sangmi Lee
Computer Science Department, Florida State University
224 Showers Bldg.
501 N. Morton St.
Bloomington, IN, U.S.A.

Abstract

We describe a collaborative environment Garnet designed around a universal event model using the publish-subscribe paradigm. We study the integration of personal digital assistants (PDA) into synchronous collaborative sessions. Our architecture uses a PDA adaptor that maps system events into a form suitable for the PDA and uses an optimized protocol that respects bandwidth limitations.

Keyword: Collaboration, Mobile, Ubiquitous, Grid, Education

1. Introduction

Universal access refers to the capability that all users are able to access information systems (grids) independent of their access device and their physical capabilities. We address this for *Personal Digital Assistants* (PDA's) and generalize it to universal collaboration – the capability of multiple users to link together over the web with disparate access modes. Mobile systems are typically slow, unreliable, and have unpredictable temporal characteristics. Further, the user interface is clearly limited. The design of distributed mobile applications needs to identify the practicalities, reliability, and possibilities of continuous interaction and integrate synchronous and asynchronous collaboration.

This paper first describes the basic PDA adaptor architecture. Our production collaboration systems use the Java Message service (JMS) [25] as their publish-subscribe engine and we describe the PDA adaptor that acts as a filter for the PDA of events from JMS. We follow this by a description of some other collaborative functions: the text chat and Instant Messenger tools, and the shared display and shared SVG export applications. We present performance measurement of Garnet Message Service (GMS) and GMS Micro Edition (GMSME) [1].

Garnet uses core components produced by the Anabas Corporation – a company of which Dr. G. C. Fox is co-founder. The work described here is independent of this commercial system.

2. Related Work

Currently, there are several collaborative systems including commercial products such as; WebEx communication Inc.'s *WebEx* [5], Centra Software Inc.'s *Centra* [6], PlaceWare Inc.'s *PlaceWare* [7], and *Latitude* [8] from Latitude Communication Inc. They provide collaborative features of Internet conferencing, collaborative meeting, and resource sharing between desktop computers. Our Garnet system is designed to be able to support both these traditional collaborative features and also access from heterogeneous user devices. One important project to adapt PDAs to work with conventional desktop computers is Carnegie Mellon University's *Pebbles PDA Project* [28]. Pebbles is designed to communicate with PDAs through a communication server, PebblesPC, and messages follow the Pebbles protocol defined in their header files. Correspondingly, the Garnet system uses a Personal Server for PDAs' communication. Moreover, Personal Servers communicate to each other via the GMS message service. The Garnet system is designed to provide scalable accessibility to heterogeneous devices for large distributed systems.

There are several other interesting approaches to collaboration between desktop computers and PDAs such as Harakan Software's PalmVNC [29], Nokia's Java VNC viewer [30], Cutting Edge Software's Quickoffice Conference [31] and FeatureBase [32] from Dawn Cooperation. PalmVNC and the Java VNC viewer are an implementation of the VNC protocol form ORL AT&T Lab, supporting shared display and events. Quickoffice is an Internet conferencing system for PDAs, and FeatureBase is Web GIS program accessible from PDAs.

3. Integration Architecture for Garnet and Hand-held device

The Garnet Collaboration system [1] is designed to support both centralized and peer-to-peer collaboration models by using a uniform event bus. This event bus is defined in XML (but implemented with a pure Java solution) and 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 clients. Note even shared display (section 5.1) is treated in this fashion with the events specifying changes in the display buffer on the master client. We have built a distributed event broker 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 termed GXOS [23].

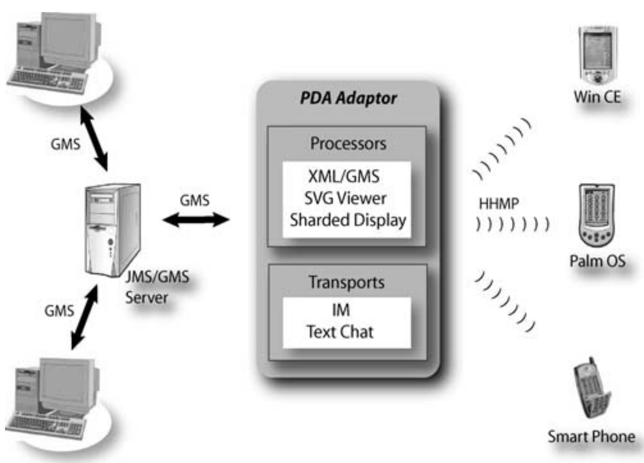

Fig 1. PDA adaptor (Personal Server).

A key additional characteristic of Garnet collaboration system treated in this paper concerns the heterogeneity of clients. Hand-held devices, mobile phones, as well as conventional desktops can join in one collaboration session. However, the conventional GMS is not able to support lightweight clients, such as PDAs because a lightweight client, mobile clients in most cases, has a smaller display and slower processor. For the integration, we propose a new universal collaboration and access architecture, which we call Garnet Message Service Micro Edition (GMSME). This supports mobile devices and desktop computers.

Garnet Message Service (GMS)	Garnet Message Service Micro Edition (GMSME)
Conventional Desktop Environment	Lightweight Clients
XML, JMS, P2P	Wireless Communication
Dynamic Routing	Optimized Protocol (HHMS)
Guarantee Delivery	User Status Management

Table 1. GMS and GMSME comparison

Some features of both GMS and GMSME are shown in Table 1. Note that Narada allows a mix of JMS and JXTA like clients with different policies for reliability of message delivery for different subsets of the clients. Thus, one can support a mix of reliable but predetermined interacting PC clients together with multiple JXTA peer groups including perhaps of hand-held devices.

GMSME consists of a PDA adaptor, a Hand Held Message Service (HHMS) protocol, and an API for application processors, which process messages on the mobile device. We describe two applications, SVG [12,15] and shared export, later. The PDA adaptor sits between the 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 "looks like" a typical Garnet client to GMS, and adapts data to the mobile device client specifications. For instance, the PDA adaptor 'listens' to all 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 message.

Just as the GMS server deals with message routing [22], the PDA adaptor of GMSME does the same task for mobile devices. Instead of the dynamic message routing server of GMS, the PDA adaptor uses a register table to manage connections and deliver messages. Of course the requirements are different for GMS and GMSME; GMS is coping with dynamic clients linked across the globe; GMSME is focused on a few localized "personal" clients which are however intermittently available and have poor communication links. The mobile user login ID, user ID, is unique over the whole collaboration system. In the GMSME system, the unique user ID is a combination of message destination address, as well as the label of client message processing unit.

HHMS is a natural modification of the event service of GMSME bearing in mind the constraints of hand held devices. The PDA adaptor parses the GMS XML message body and delivers it to the mobile client over HHMS. As already noticed, mobile clients have modest performance and size in comparison with classic desktop machine. Therefore, they require particularly efficient protocols. The HHMS protocol is efficient and not very elaborate. The structure of HHMS is shown in Fig 2. Like GMS, HHMS also sends event and data between software applications. In the HHMS case, software applications are Garnet collaboration system and mobile applications. JMS-based GMS messages contain many properties, such as a destination, a delivery mode, a messageID, which can be extracted from a message header [25], as well as a complex message body. GMSME does not use 'dynamic routing' as supported by GMS. Thus, many of GMS message properties are not necessary for GMSME and mobile client. We select the essential fields and map

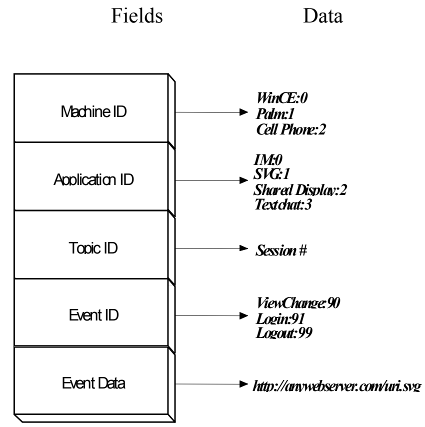

Fig 2. Anatomy of a HHMS Message

them to a binary message. Consequently, HHMS protocol contains only the necessary data for message delivery in optimized form. Several GMS message fields are implemented in different ways. For instance, the GMS message priority is implemented in GMSME by manipulating a job queue. When an emergency message needs to be delivered, it is just placed at the head of the queue. The PDA adaptor keeps data in the form of a hashtable, to enable priority changes.

GMS over JMS	HHMS of GMSME
Publish message	SendHHMS and SendGMS. Mapping the messages between GMS system and mobile clients.
Subscribe message	Register and login
MessageQueue	JobQueue keeps the load inside limits
Recovery	Logfile records major events

Table 2. GMS and HHMS Messages

GMSME supports heterogeneous clients by customizing message depending on the client specifications. We keep user profile and mobile device profile, such as screen size and device type in the PDA adaptor. 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 from the conventional Garnet collaboration system with information on the mobile device. In addition, these profiles make it possible to render messages differently for the mobile client. 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 modification to provide personalized and optimized look.

There are some key design issues for GMSME to provide mobile devices the same level of service as desktop computers. We need to achieve 'optimized performance', 'extensible systems' and 'robust systems'

Let us consider now performance issues for GMSME, where since the performance of desktop computer exceeds that of mobile device, efficiency is very critical. We accomplish this optimization by two design strategies:

- *Move process to server side:* To reduce processing 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. Because of this preprocessing, the mobile client receives a ready to use image from the PDA adaptor.
- *Optimize protocol, HHMS:* HHMS is byte oriented message protocol between mobile client and the PDA adaptor on desktop. Tag bytes in the byte array encode application types, event types, and event message.

We consider two extensibility requirements. The first is a standard interface (API) to the processing module and to the mobile client. All information passing between the PDA adaptor and processing modules uses a *hashtable* object of Java and so additional modules can be easily added. For the communication interface to the mobile client, we have a standard Java interface to deliver byte arrays to and from the PDA adaptor. Currently, we use various communication methods between mobile devices and the PDA adaptor. 802.11b wireless LAN [24] is used as an initial communication media but we are also developing support for standard cell phone and PDA data networks. Despite some security issues, 802.11b provides high bandwidth. For cellular phones and most cellular communication using hand held devices, HTTP is the most popular protocol. Because of characteristic pull architecture of the HTTP protocol, we use a polling mechanism to maintain connection with the mobile device. Mobile devices make frequent requests to retrieve messages from the PDA adaptor.

4. Instant Messenger and Text Chat on PDA

We have integrated PDA based Instant Messenger (IM) and Text Chat into the GMS publish/subscribe collaboration environment. We base this capability on the popular *Jabber Open server* [21] using XML presence protocol (XXMP) and GMS using the GXOS XML framework for messages. The benefits of using the Jabber Open server include presence management, message processing based on XML, transparent interoperability, structured information data, and open formats. The Jabber server can link to other messenger services like ICQ, MSN, and Yahoo.

The IM on the iPaq PDA was built using "PersonalJava" environment. It connects first to the PDA adaptor using TCP connection and HHMS protocol. The IM transport shown in fig. 3 runs on the PDA adaptor and establishes a connection to Jabber Open server. Each IM user on an iPaq can send messages to other IM users and check online or offline presences of IM users in real time. On the other hand, the Text Chat on iPaq connects to the PDA adaptor in a similar way but now its transport on the adaptor publishes messages to the GMS publish/subscribe server.

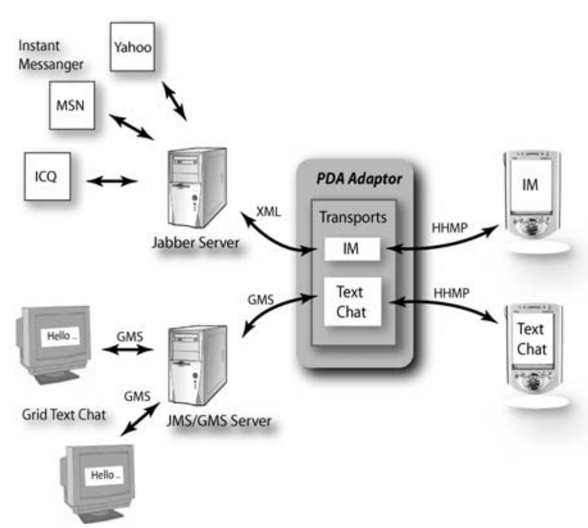

Fig 3. IM and Text Chat Transports

Note that currently we have not integrated the Instant Messenger system into Garnet although this was available in an early prototype. One can either add IM capability to Garnet or modify the Garnet text chat to interact with IM on PDA.

5. Shared applications on PDA

Here we discuss object sharing in real-time with events recording state changes transmitted from a "master" instantiation to replicas on other clients in the same session. The documents include visualizations, web pages and Microsoft Word documents and are shared with shared event, shared export or shared display mechanisms on the PDA.

5.1. Shared Display

Shared Display is the simplest method for sharing documents with the frame-buffer corresponding to either a window or entire desktop replicated among the clients. Modest client dependence is possible with PDA's for example receiving a reduced size image. Some collaboration systems support remote manipulation with user interactions on one machine holding a replica frame-buffer transmitted to the instance holding the original object. This is an important capability in help desk or consulting applications, similar to situations that occur frequently in the debugging of code. As shared display works for all applications without modifying them, this is the basic shared document mechanism in Garnet System [1].

Garnet sends GMS events containing an encoded copy of the part of the framebuffer that has been updated. To cope with "late clients" and fault tolerance, major events containing the full frame buffer are sent at regular intervals. The encoding method in Garnet system is adaptive. For instance, if *Huffman* and *LZ77* are requested but the Huffman compressor does not reduce the data, the Huffman request will be skipped and LZ77 will be just processed.

Garnet's shared display is accessible from small mobile devices through the PDA adaptor that resizes the display for small devices, which have low resolution [2]. The PDA adaptor requests updated data periodically, and optimized data is transferred to PDA as raw bitmap data with HHMS. We developed a PDA client application in the PersonalJava runtime environment for Windows CE. This client application is initially designed for Compaq Inc.'s *iPAQ* [9], which has a display resolution of 320 x 240.

5.2. Shared Export

Shared display does not allow significant flexibility; for instance, different clients cannot examine separate viewpoints of a scientific visualization. More flexible sharing is possible by sharing object state updates among clients with each client 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.

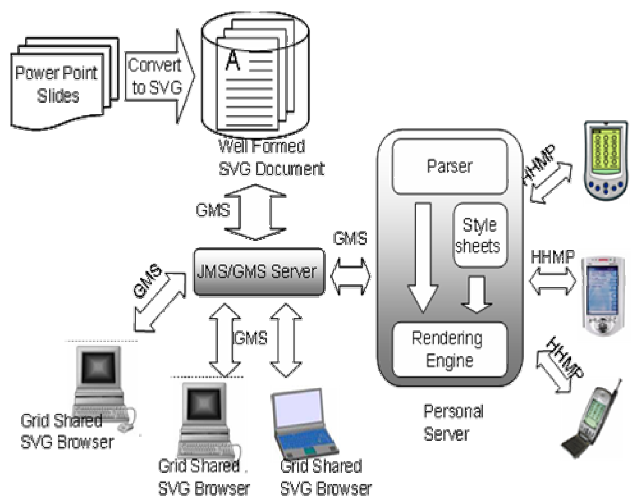

Fig 4. Architecture of Collaborative Export with SVG

The scalable formats of 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]. W3C's *Annotea* [17] allows similar general collaborative viewers for desktop computers to support collaborative annotations, which are anchored to objects within the SVG document. Meanwhile, Garnet system supports collaborative annotation at general pixel positions, which are then rendered at appropriate scaled position on each client synchronously.

An exported SVG document in the Garnet System is shared by broadcasting its URI and the PDA adaptor subscribes to this information. Each PDA adaptor renders the image appropriately for different types of wireless devices. We define SVG elements for low-resolution devices with Cascading Style Sheets (CSS) on the PDA adaptor. Display types of small wireless devices are various: from 16bits color with 320x320 resolution [18] to 1 bit gray scale with 160x160 resolution [19]. Every image rendered by PDA adaptor is transferred to each device as a JPEG image format. We have developed the integrated client application for Compaq Inc.'s iPAQ, which displays the JPEG image transmitted by HHMS from the PDA adaptor.

The Garnet System provides the interactive framework for collaborative SVG export. As a basic function, our system supports collaborative annotation. Each event for the annotation is translated to SVG document and sent to other participants as a message object with GMS service. The received message is parsed and rendered dynamically. The SVG event processing is based on the W3C document model. Each exported SVG document creates DOM tree, any additional item is added to the DOM tree with DOM interface, and rendered dynamically. Every event message is encoded with XML syntax and delivered to other participants via GMS.

6. System Performance

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 are not preserved in a case of any system failure. Multi-user communication and shared application performance will be evaluated in future work.

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 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 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

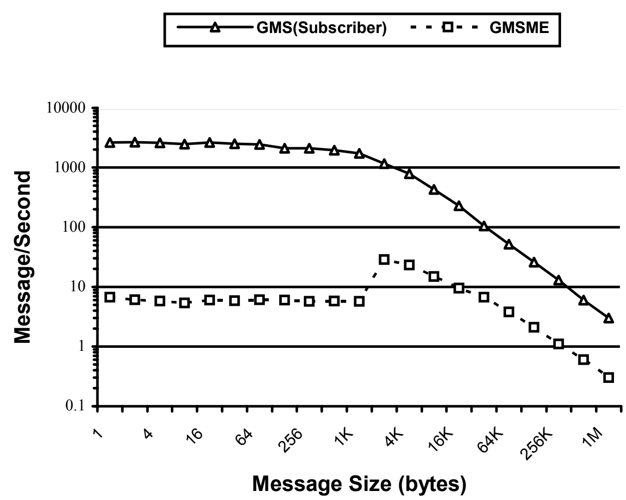

Fig 5. GMS and GMSME Message Rate.

Figure 5 shows the average message throughput per second for different message size. It is clear that GMSME is significantly slower than

GMS 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 4K. This indicates that we would expect best performance for PDA Shared Display when an image is divided into 4K 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.

7. Summary and Future Work

GMS guarantees message delivery but we adopt a different strategy in GMSME. Garnet identifies "major events" which are those from which one can restart a shared application. Thus, we use a log file listing all major events in the PDA adaptor for each user in GMSME. It allows the PDA adaptor to be able to recover the state of each user from the initial connection or unexpected disconnection from collaboration system. Moreover, we introduced a job queue in the PDA adaptor. The job queue in the PDA adaptor keeps 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 tolerance approach for PDA adaptor itself. This is delegated to GMS, which guarantees to archive and deliver events. So if an adaptor fails, we switch to an alternative adaptor host, which can restart the PDA collaboration. This could preserve critical information such as adaptor log files as long as the original adaptor publishes these on the GMS publish/subscribe server.

We will extend this work to other mobile devices such as cellular phones using J2ME (Java 2, Micro Edition). We can also add SMS (Short Message Service) on mobile devices for sending brief messages between devices like cell phones by using SMS-Jabber Open server currently in the planning phase and SMTP (Simple Mail Transfer Protocol) service on mobile devices for electronic mail used by most email services. In a different project we have already linked SMTP XML (SOAP) transport with GMS publish/subscribe environment. Using SMS transport and SMTP transport to exchange messages between disparate platforms into the PDA adaptor, we can support disparate mobile information devices for universal collaboration and access.

References

[1] G. C. Fox. *Architecture and Implementation of a Collaborative Computing and EducationPortal.* 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/erdcpda.pdf

[3] AT&T Laboratories, *VNC - Virtual Network Computing from AT&T Laboratories Cambridge*, http://www.uk.research.att.com/vnc/

[4] Microsoft, Inc. *NetMeeting: Microsoft commercial product description from http://www*.microsoft.com/netmeeting

[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] Latitude Communication, Inc. *Latitude*. http://www.latitude.com

[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.cs.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] Sony, Inc. *CLIE Personal Entertainment Organizer Specifications*, http://www.sonystyle.com/micros/clie/760specifications/PEG-N760C.html
[19] Palm, Inc. *Palm m500 Handheld Product Details*, http://www.palm.com/products/palmm500/details.html
[20] Sun Microsystems Inc. *Personal Java 1.0 Specification*. http://java.sun.com/products/personaljava/spec-1-0-0/personalJavaSpec.html
[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]The Institute of Electrical and Electronics Engineers, Inc., *IEEE 802.11b Local Area Network* http://grouper.ieee.org/groups/802/11/
[25] R. Monson-Haefel and Chappell, D.A., *JAVA Message Service*, 2001.
[26] G. C. Fox, "Peer-To-Peer Networks", *Computing in Science & Engineering*, vol. 3, no 3, May 2001
[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. November, 2001. pp. 34 - 41
[29] Harakan Software, *PalmVNC*, http://www.btinternet.com/~harakan/PalmVNC/
[30] Nokia, *Java VNC viewer*, http://www.mgroeber.de/nokia.htm
[31] Cutting Edge Software, *Quickoffice Conference*, http://www.cesinc.com/products.
[32] Dawn corp., *Feature Base*, http://www.dawn-corp.co.jp/gis/index.html