Universal Accessible Collaboration Frameworks for Ubiquitous Computing Environments

Geoffrey Fox[1,2], Sunghoon Ko[1], Kangseok Kim[1,2], Sangmi Lee[1,3], and Sangyoon Oh[1,2]
Community Grid Computing Labs, Indiana University [1]
Department of Computer Science, Indiana University [2]
Department of Computer Science, Florida State University[3]
{gcf, suko}@indiana.edu, slee@csit.fsu.edu, {kakim, ohsangy}@cs.indiana.edu

Abstract

The advances in mobile devices, wireless communication, and messaging technologies have given rise to a vast range of new services, for heterogeneous user environments. These include a collaboration service that can be accessed regardless of the nature of user devices and physical capabilities of the user. In this paper, we present an overview of our research project *CAROUSEL,* universal accessible collaboration frameworks. We describe two collaborative frameworks; our initial approach using a special PDA adaptor and current investigation using Web Services model. Although the two systems are functionally similar, there are differences in the degree of interoperability, scalability and reusability of the infrastructure. In Web Services case, we discuss how a rich synchronous and asynchronous collaboration environment can support virtual communications while built on top of a Web service infrastructure. We also describe the approach to universal access mechanisms in our collaborative model.

1 Introduction

The CAROUSEL [11] project carries out research on the infrastructure for the collaboration system supporting universal access. Collaborative service provides the capability for geographically distributed users to share resources and work together on a specific problem. *Universal access* refers to the capability that all users are able to access information systems independent of their access device and their physical capabilities. With the emergence of various portable devices and advances in wireless network communications, current collaborative systems require universal accessibility. We address the universal access problem for PDAs (Personal Digital Assistants) 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. Synchronous and asynchronous collaboration both involve object sharing but the former is fault sensitive, has modest real-time constraints and requires fine grain object state synchronization. A similar sharing mechanism can be used in our approach for both synchronous and asynchronous cases. One needs to establish communities (peer groups) by either direct methods (members join a session) or indirectly (members express interest in topics and are given opportunity to satisfy this interest). The indirect mechanism is most powerful and is familiar in P2P [24] systems with JXTA [21] using XML expressed advertisements to link together those interested in a particular topic.

Collaboration involves multiple capabilities which include

- Mechanism to set up members (people, devices) of a "collaborative sessions" and their properties
- Generic shared tools such as text chat, white boards, and audio-video conferencing
- Shared applications such as Web Pages, PowerPoint, and scientific visualization

In this paper, we describe two frameworks that support universal access in collaboration environment. We developed first case with a special adaptor that is essentially a client (node) to event-brokering. This adaptor has filters controlled by client profile and special purpose messaging protocol HHMS (Hand Held Message Service) [20] optimized for PDA. Currently, we are investigating new collaborative framework using Web Services model. This Collaborative Web Services gives the interoperability between services, scalability of system and reusability of service modules. In this framework, one of the steps to enable universal accessibility is by intelligently defining user "profiles" and the semantics of Web services. We will describe how collaboration and universal access can be incorporated in the collaborative Web Services architecture. The linkage of the event brokering system with a Web Services based architecture is one of the critical design issues in message-based collaborative application.

The paper starts with our general approach for universal collaboration. In section 3, we present our first approach to a collaboration framework that supports hand-held devices. Section 4 introduces our current investigation of collaboration system based on Web Services model. Finally we conclude in section 5.

2 Methodology and Approach

While adapting new devices, one should consider various factors: network communication mechanisms, computing power, display capability, etc. In this section we present some strategies that can be applied to collaboration services for hand held devices. We want to support the convenient interaction between multiple resources and users in heterogeneous environments.

Content adaptation

Transcoding technology helps enable universal access by dynamically adapting, reformatting and filtering Web content and applications to make them optimally suited for limited capable mobile devices, such as phones, PDAs and other pervasive devices. This allows creating content once and then reformatting it on the fly for delivery to a range of gadgets. In this project, we utilize various transcoding technologies to adapt shared content based on the type of collaboration [22]. Here transcoding technology includes image resizing, converting image formats, compressing data, and transform technology such as the use of stylesheets. There have been a number of approaches to accommodate mobile devices by transcoding technologies [5-9]. However, different collaborative features may need different transcoding technologies according to their unique functionality. Transcoding can be implemented either as an insertion of a filter in the PDA message stream or in batch (although batch delay can be as small as a millisecond) mode where a service subscribes to event stream (one collaborative application), filters it and reposts to a different stream. CAROUSEL uses transcoding technology to support two different and popular resource sharing model; *shared display* and *shared export*. In shared display, one shares the bitmap display and the state is maintained between the clients by transmitting the changes in the display. Meanwhile, 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. Document formats such as W3C's Scalable Vector Graphics (SVG) [15] or Adobe, Inc.'s Portable Document Format (PDF) are particularly interesting and support of collaborative viewers is a major advantage of our system. With the scalability of vector graphics, and separation of user presentation from the master's content, shared export provides more flexibility for scientific visualization or geographical information systems.

Performance issues

Since the performance of desktop computer exceeds that of mobile device, efficiency is very critical. We accomplish this optimization by moving processes to the server side to reduce the compute load on the mobile device. We move part or all of the processing module for each application from the hand held device to the PDA adaptor on the server side. This also helps customize content delivery and presentation with device, network and end user profiles to enable more effective interaction with mobile users. 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 processing, the mobile client receives a ready to use image from the PDA adaptor. In addition, optimized protocol is required for mobile services. Our HHMS is optimized 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. The essential fields are selected and mapped to a binary message. Consequently, HHMS protocol contains only the necessary data for message delivery in optimized form.

Event service

The CAROUSEL collaboration services use message-based systems to offer rich synchronous/asynchronous collaboration features. To provide messaging between the heterogeneous user network environments and Web services, NaradaBrokering [10] from the Community Grid Labs is adapted as a general event brokering system. NaradaBrokering supports centralized, distributed and peer-to-peer (P2P) messaging models with a dynamic collection of brokers supporting a generalized publish-subscribe mechanism. NaradaBrokering can operate either in a client-server mode like JMS (Java Message Service) [3] or in a completely distributed JXTA-like peer-to-peer mode [21]. By combining these two disparate models, NaradaBrokering can allow optimized performance-functionality trade-offs for different scenarios. At the transport level NaradaBrokering provides support for TCP, UDP, Multicast, SSL and RTP. For remote resources behind a firewall, NaradaBrokering provides the capability to communicate across firewall/proxy boundaries.

For mobile users, such as PDAs and smart phone, the HHMS protocol [20] will be integrated as a communication protocol within the NaradaBrokering transport framework. This would allow PDA's and other devices to interact directly with the messaging system; and with each other across firewalls, proxies and NAT boundaries. We expect that the collaborative system developed based on NaradaBrokering's messaging infrastructure will provide the collaboration service to the users in heterogeneous network environments with greater reliability within a scalable network framework.

In the next two sections, we describe two collaboration frameworks based on the above methods. Though the two systems provide same features, there are differences in the degree of interoperability, scalability and reusability.

3 The PDA Adaptor Approach

The Garnet Collaboration system [12] is our first effort to universal accessible collaboration framework that designed to support both centralized and peer-to-peer collaboration models by using a uniform event bus. This event bus is defined in XML 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. This system initially used JMS (Java Message Service) but now is built on a NaradaBrokering event broker system. The resultant event service is called the GMS (Garnet Message Service) whose XML specification is termed GXOS [13].

Integration of PDAs using Adaptor

A key additional characteristic of Garnet collaboration system 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 proposed a new universal collaboration and access architecture, which we call Garnet Message Service Micro Edition (GMSME) [20]. This supports mobile devices and desktop computers. Figure 1 shows collaboration system with GMSME. 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. 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 [10], the PDA adaptor of GMSME does the same task for mobile devices. 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.

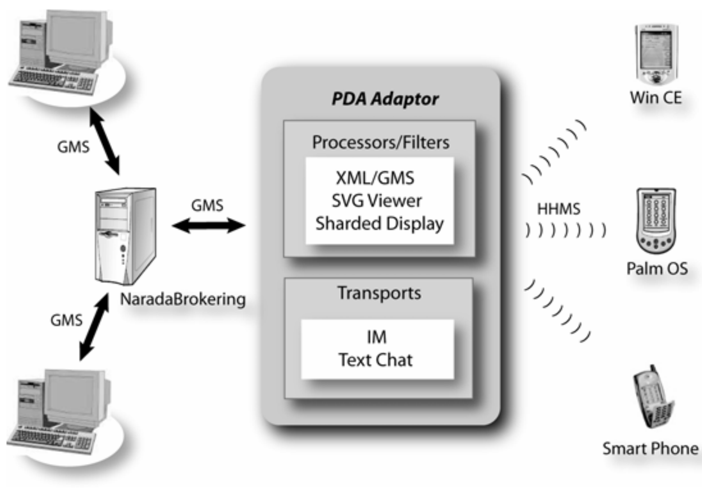

Figure 1. Collaboration system using PDA Adaptor

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. 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, 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 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.

GMSME supports heterogeneous clients by customizing message depending on the client specifications. We keep user profiles and mobile device profiles, containing information 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 [36]. 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'

Currently, we use various communication methods between mobile devices and the PDA adaptor. 802.11b wireless LAN is used as an initial communication media but we have also developed initial 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.

One key advantage of this design is that it can make a system provide mobile service by simply adding PDA adaptor on the pre-existing system that does not support service for hand held devices. However it lacks interoperability, scalability and reusability. Different implementation and integration are required to provide same features on different system. For flexible and inexpensive integration capabilities, we introduce collaboration framework using a service-oriented architecture (SOA) [16] based on Web Services. This Collaborative Web Services gives the interoperability between services, scalability of system and reusability of each service module. In this framework, one of the steps to enable universal accessibility corresponds to intelligently defining user "profile" and the semantics of Web Services. In the next section, we will describe how collaboration and universal access can be incorporated in the collaborative Web Services architecture.

4 The Web Services Approach

The Web Services concept allows objects to be distributed across web sites where clients can access them via the Internet. Moreover the use of XML [14] and standard interfaces like WSDL (Web Services Definition Language) [2] gives the interoperability between services, including those in Grids [19], which are formed from robust largely asynchronous shared resources. At a high level, collaboration involves sharing and in our context this is the sharing of Web Services, objects or resources. We can expect that Web Services interfaces to "many software packages" will be available and we will take this point of view below; where MS-Word, a Web Page, computer visualization or the audio-video (at say 30 frames per second) from some video-conferencing system will all be viewed as objects or resources with a known Web Services interface.

The current Web Services infrastructures such as Apache project's Jetspeed [4] or WebSphere [17] from IBM provide services and development tools for mobile devices. However, these approaches based on portal services are not efficient enough to support real time services such as synchronous collaborative services. Current multi-functional collaboration services require a scalable framework to include new Web Services and third-party Web applications. The research pertaining to scalability will be focused on how we define needed functionalities as Web Services, with ports providing interoperability between these services. Web Services are started with the exchange of *messages* between the service provider and the requestor. WSDL defines the exchange of messages between the service provider and the requestor as an *operation* [2]. The messages in operations are described abstractly, and bound to a concrete network protocol and message format. The operation is defined with an input or output port as a minimal description. Our research extends negotiation capabilities on output ports to support diverse dynamic client requirements for universal access.

This system will be an advanced research prototype of the "sharing input/output port" [1] collaboration model. Our approach originates from the refinement of this data processing. Each object is processed in a well-defined data flow called the "data-pipeline", which is designed to facilitate flexible data management. Each stage of the pipeline is a Web Service with data flowing from one stage to another. CAROUSEL Web Services are designed in terms of the concept of "data-pipeline" and provides the infrastructure integrating multiple resources for heterogeneous users in a ubiquitous environment [23].

4.1 Dataflow and Collaboration

Here we discuss the concept of dataflow and the collaborative service model utilized in the CAROUSEL Web Services. With regards to an object in Web Services, the object is typically in some pipeline, as seen in figure 2(a), from the original object to the eventual displayed user interface. Each stage of the pipeline is a Web Service with data flowing from one stage to another. We consider the output of each stage of this pipeline as a "document" – each with its own document object model; preferably different instances of the W3C DOM (Document Object Model). The final user interface could be a pure audio rendering for a visually challenged user, or a bitmap transmitted to a primitive client not able to perform full browser functions.

In order to describe a general approach to collaboration, we need to assume that every Web service has one or more ports in each of the three classes shown in figure 2(b). The first class is *resource-facing input ports*, which supply the information needed to define the state of the Web service. *User-facing input port*(s), which allow control information to be passed by the user, may augment these resource-facing input ports. The final class is *user-facing output ports* that supply information needed to construct the user interface. Asynchronous collaboration can share the data (e.g. URL for a Web page or body of an email message) needed to define a Web service (display Web page or browse e-mail in examples).

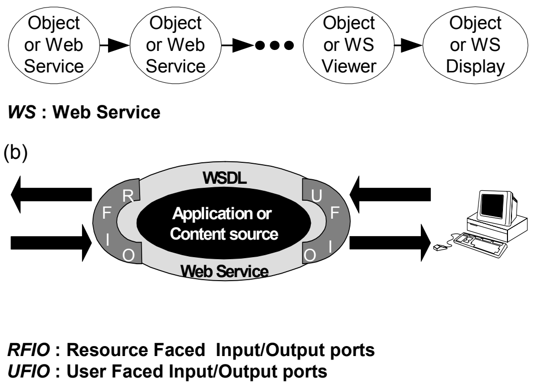

Figure 2. (a) Web Services pipeline flow from originating objects to display.

(b) Web Services can have resource and user facing ports.

In the CAROUSEL Web service, each collaborative application can get different types of sharing depending on which "view" of the basic object one shares i.e. on where one intercepts the pipeline and shares the flow of information after this interception. We can identify three cases; these are shared display, shared input port, and shared user-facing output ports [1]. Sharing either the input or user-facing ports of a Web Services allows one to build flexible environments supporting either the synchronous or asynchronous collaboration needed to support the communities built around this infrastructure. In this paper we focus on an overall architecture and universal accessibility.

4.2 The Linkage with Event Brokering System

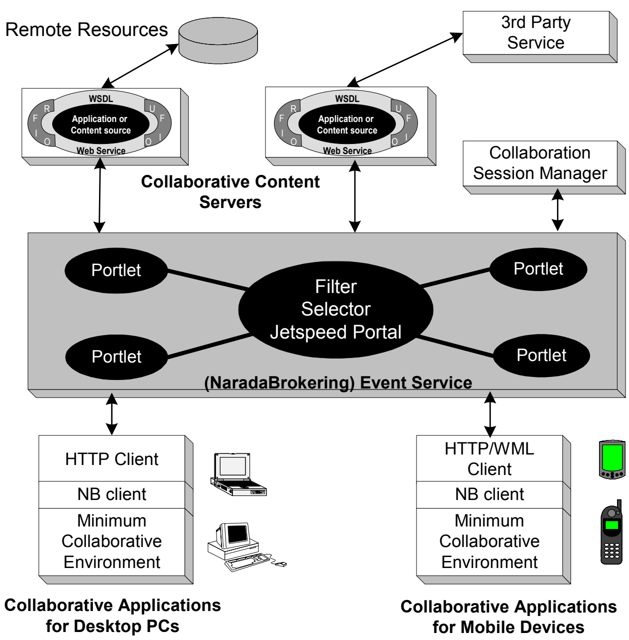

Figure 3. CAROUSEL Web Services Architecture

The CAROUSEL Web service contains four major components: content servers, aggregator, client application and event service. The set of cooperating services in our framework are deployed using the message service for communication. Each component is linked together with their input/output ports designed to support Web services semantics.

Each collaborative feature of the CAROUSEL Web service is designed with individual remote content servers shown in figure 3. The content server is designed with a portlet Web service user interface, which cover user-facing input and output ports. The overall structure of every message, including input and output, are defined in WSDL[2]. The output/input ports support dynamic communication channels to NaradaBrokering and general HTTP communication.

The client application is designed to require minimal data processing. The customized output data is delivered and displayed through the client application. The major features of the client application are; user specification, display portlet presentation, and processing user input events.

Several collaborative features designed as content servers and supporting services, are implemented as portlets and can be aggregated within a portal. The user sees the portal presentation view, which is the aggregation of the presentations of each portlet. The portlet presentation views can be in separate windows and can either be distinct or partially layered on top of each other. This portlet presentation view includes all kinds of machine-dependent outputs such as bitmap display or audio streams.

In the next section, we clarify some user interface issues which we discuss in the context of universal accessibility.

4.3 Universal Access in CAROUSEL Web service

Universal access in the CAROUSEL Web service is approached with the user output/input defined intelligently by an interaction between the user "profile" (specifying user and client capabilities and preferences) and the semantics of general Web services [4]. The service itself specifies the most important parts of its user-facing view and the output should be modified for clients with limited capabilities.

In WSDL, the inputs and outputs of operations are termed as *ports*. The CAROUSEL Web service is designed with one or more ports in each Web service component to provide a general approach as collaborative application. Here we will discuss how we linked each Web service component via these input and output ports and approach the universal access in this object flow of Web services.

Each Web service is designed with three major user-facing ports as an output port of the modular pipeline of figure 4. First, there are the main user-facing specification output ports that in general do not deliver the information defining the

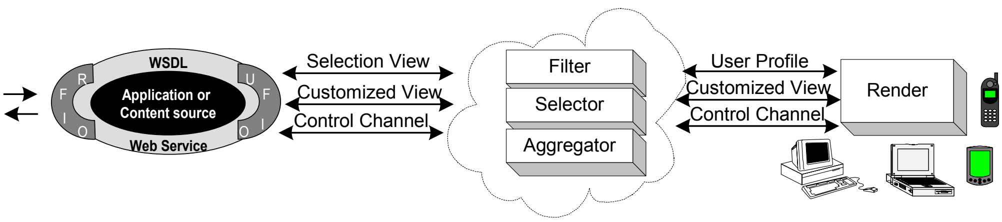

Figure 4. Architecture of Event Service and Portal to support Universal Access

display, but rather a menu that defines many possible views. A selector in figure 4 combines a user profile from the client (specified on a special profile port) with this menu to produce the "specification of actual user output" that is used by a *portal*, which aggregates many user interface components (from different Web services) into a single view. The result of the transformer may just be a handle, which points to a user facing customized output port. This output port allows users to select user interface components - operating systems, display types, and resolution preferences.

Second, the customized user-facing output port that delivers the selected view from selector of the Web service to the client. This in general need not pass through the portal, as this only needs the specification of the interface and not the data defining the interface.

Finally, User-facing input/output port, which is the control channel, shown in figure 4.

Note that in figure 4, we have lumped a portal (such as Jetspeed [4] from Apache) as part of the "event service" as it provides a general service (aggregating user interface components) for all applications (Web services). This packaging may not be the most convenient, but architecturally, portals share features with workflow, filters and collaboration. These are services that operate on message streams produced by Web services. Considering universal access in this fashion could make it easier to provide better customizable interfaces and help those for whom the current display is unsuitable.

5 Conclusion and Future work

We introduced two architectures for providing universal accessibility to Web based collaborative services. We describe our first approach with a PDA adaptor which is specially designed for access to general purpose event brokering system (NaradaBrokering). An adaptor has filters controlled by client profile and special purpose link protocol HHMS (Hand Held Message Service) optimized for PDA. Currently system supports for Windows CE iPaq and Palm OS Cell-PDA combination as shown in figure 5. This model lacks interoperability, scalability and reusability. Different implementation and integration are required to provide same features on different system. For flexible and inexpensive integration capabilities, we are currently engaged in the collaborative Web Services based on the sharing either the input or user-facing ports of a Web Services for more flexible collaboration. Each Web service in this architecture provides one or more ports facing to resources and users to implement an integrated collaboration application. Both frameworks are linked to a message based event service, and every message is defined with XML. We discussed object-flow pipeline in CAROUSEL Web service, and how we deploy it to facilitate the universal access.

Universal access requires that the user interfaces be defined intelligently by an interaction between the user "profile" and the semantics of general Web services. Only the service itself specifies the most important parts of its user-facing

Figure 5. Heterogeneous Collaboration sharing PowerPoint presentation.

view and the output should be modified for clients with limited capabilities. This implies that the data processing in service is deficient in the sense that there must be a clear flow not only from the "basic Web services" to the user, but also back again. This can be quite complicated and it is not clear how this is achieved in general as the pipeline from Web services to user can include transformations, which are not reversible. For this reversibility problem, the content servers of the CAROUSEL Web service are designed so that they keeps the original document internally, and provide an interface to the original document to generate new output for each event. The ambiguity of reverse functionalities still exists, but we can expect that every reverse function will get correct output with this design.

For future work, we will integrate the HHMS protocol as a communication protocol within the NaradaBrokering transport framework. We expect that it will provide more seamless network communication to mobile users collaborating with users in traditional computers. We are also investigating more reliable communication methods for mobile devices accessing the CAROUSEL Web service. Integration of 3rd party services into the Web service architecture is another issue that is under investigation. This includes wrapping and integrating remote service from other venders, such as the XML Presence Protocol based instant messenger (XXMP) [18] from Jabber Open server. The research on the mobile communication protocols continues in parallel. With the remarkable improvements in mobile services, there are more possibilities to adapt new communication technologies for PDAs such as JXTA and JMS style event brokers. We expect that the investigation of these communication methods will provide better performance and quality of service.

References

[1] Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko, Sangmi Lee, Shrideep Pallickara, Marlon Pierce, Xiaohong Qiu, Xi Rao, Ahmet Uyar, Minjun Wang, Wenjun Wu Book chapter on Peer-to-Peer Grids http://grids.ucs.indiana.edu/ptliupages/publications/p2pgridbook.pdf
[2] Web Services Description Language (WSDL) version 1.1 http://www.w3.org/TR/wsdl
[3] Sun Micro Systems, Java Message Service http://java.sun.com/products/jms
[4] Jetspeed Portal from Apache http://jakarta.apache.org/jetspeed/site/index.html
[5] S. Chandra, A. Gehani, C. Ellis and A. Vahdat, Transcoding Characteristics of Web Images, Multimedia Computing and Networking, Jan, 2001, San Jose, CA.
[6] Intel Inc., Intel quickweb, http://www-us-east.intel.com/quickweb/
[7] Oracle Inc., "Oracle portal-to-go; any service to any device", white paper, Oct,1999, http://www.oracle.com/mobile/portaltogo/
[8] IBM, "Websphere transcoding publisher.", 2001, http://www.ibm.com/software/webservers/transcoding.
[9]J.R. Smith, R. Mohan, and C.Li, "Scalable multimedia delivery for pervasive computing." In Proc. of ACM Multimedia, Oct. 1999, pp.131-140, ACM Multimedia, Orlando, Florida
[10] 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/
[11] CAROUSEL project, http://grids.ucs.indiana.edu/ptliupages/projects/carousel/
[12] 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
[13] PTLIU Labs GXOS, http://aspen.ucs.indiana.edu/~balsoy/gxos
[14] W3C, Extensible Markup Language (XML), http://www.w3.org/XML/
[15] W3C, Scalable Vector Graphics (SVG), http://www.w3.org/Graphics/SVG/Overview.htm8
[16] Service-Oriented Architecture (SOA) http://www.service-architecture.com/
[17] IBM, WebSpheres, http://www-3.ibm.com/software/info1/websphere/index.jsp
[18] P. Saint-Andre, Jabber Server v1.2 Technical White Paper, Jabber.com, Inc., http://www.jabber.com/pdf/Jabber_Server_White_Paper.pdf
[19] The Grid Forum http://www.gridforum.org
[20] Geoffrey Fox, Sung-Hoon Ko, Kangseok Kim, Sangyoon Oh, Sangmi Lee, "Integration of Hand-Held Devices into Collaborative Environments "; proceedings of the 2002 International Conference on Internet Computing (IC-02) , June 24-27 Las Vegas. http://grids.ucs.indiana.edu/ptliupages/publications/PDA_IC2002.pdf
[21] Sun Micro Systems JXTA Peer to Peer technology. http://www.jxta.org
[22] Sangmi Lee, Sunghoon Ko, Geoffrey Fox, "Adapting Content for Mobile Devices in Heterogeneous Collaboration Environments" in Proceedings of the 2003 International Conference on Wireless Networks Las Vegas June 2003
[23] Sangmi Lee, Sunghoon Ko, Geoffrey Fox, Kangseok Kim, Sangyoon Oh "A Web Service Approach to Universal Accessibility in Collaboration Services" in Proceedings of 1st International Conference on Web Services Las Vegas June 2003
[24] PEER-TO-PEER, edited by Andy Oram, O'REILLY, 2001