

CAROUSEL Web service: Universal Accessible Web service Architecture for Collaborative Application

Geoffrey Fox¹, Sangmi Lee², Sunghoon Ko³, Kangseok Kim⁴, Sangyoon Oh⁵

Community Grid Laboratory Indiana University Bloomington, IN, USA {gcf ¹ , <a href="mailto:shko<sup>3</sup>@indiana.edu">shko³@indiana.edu	Department of Computer Science Florida State University, USA <a href="mailto:slee<sup>2</sup>@csit.fsu.edu">slee²@csit.fsu.edu	Department of Computer Science Indiana University, USA {kakim ⁴ , ohsangy ⁵ }@cs.indiana.edu
---	---	--

Abstract

We describe collaboration models in Web services, and how to achieve universal access in the heterogeneous devices including conventional desktop PCs and mobile devices. We discuss how a rich synchronous and asynchronous collaboration environment can support virtual communications built on top of Web service infrastructure based on industry standard portal technologies, such as XML, Apache project's Jetspeed, and WSDL. The linkage of the event brokering system with Web service architecture is one of the critical design issues in our message-based collaborative application. We focus on approaches to universal access mechanism and modular data pipeline in Web services to show how we implement the universal accessibility in collaborative Web service.

1. Introduction

Although Web Services are very loosely defined, the Web Service concept allows objects to be distributed across web sites where clients can access them via the Internet. Moreover the use of XML and standard interfaces like WSDL (Web service Definition Language) [2] gives a universality that allows the interoperability between services from different sources.

Local or remote Web services are integrated into portals as portlets which are user-facing, interactive web application components. Web service is started with the exchange of *messages* between the service provider and requestor. WSDL defines an exchange of messages between the service provider and 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 input or output port as a minimal description. A collection of operations is called a *portType* and they are grouped as a *serviceType*. Each *serviceType* is implemented as a *service* which contains a collection of ports. Meanwhile, the port is an implementation of a *portType* and includes all the concrete details needed to interact with the service.

We developed architecture for collaborative system supporting universal access in Web services

environment: CAROUSEL Web service. CAROUSEL Web service is general purpose collaboration Web service architecture supporting heterogeneous devices. Each Web services of CAROUSEL Web service are aggregated with the industry standard portal technology, such as Apache project's JetSpeed [4]. Jetspeed is an Enterprise Information Portal using Java and XML, which is accessible via a web browser, WAP-phone, pager or any other device. In CAROUSEL Web service, Jetspeed acts as the central hub where information from multiple Web services, and provides user customizable portal presentation view.

We will discuss how we design major components of CAROUSEL Web service, and integrate them together in sec. 3. We approach the universal access with the user interface defined intelligently by an interaction between the user "profile" and the semantic of Web services. In sec.4, we describe how we defined the modular dataflow in our design and linked the stages considering universal access.

Currently, the sophisticated collaboration system is designed over the message-based architecture because of the powerful capability of messaging systems. Besides the fast event transmitting, reliable and versatile functionalities of modern messaging services are critical to implement synchronous and asynchronous collaboration features. CAROUSEL Web service is designed to link to distributed event service, NaradaBrokering [11] which team of the Community Grid Lab led by Shrideep Pallickara developed.

2. Related Works and background of project CAROUSEL

Currently, there are several successful collaborative systems including commercial products; 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. Garnet system is designed to be able to support the traditional collaborative features and access from the heterogeneous user devices. The one of the early efforts to adapt PDAs to work with conventional desktop computers is Carnegie Mellon University's *Pittsburg Pebbles PDA Project* [13]. Pebbles is designed to communicate with PDAs through the communication server, PebblesPC, and every message is conforms to the Pebbles protocol defined in their header files.

There are other interesting approaches for collaboration between desktop computers and PDAs such as Harakan Software's PalmVNC [14,19], Nokia's Java VNC viewer [15,19], Cutting Edge Software's Quickoffice Conference [16] and FeatureBase [17] from Dawn Cooperation. PalmVNC and Java VNC viewer is an implementation of the VNC (Virtual Network Computing) protocol [19] form ORL AT&T Lab, supporting shared display and events. Quickoffice is an Internet conferencing system for PDAs.

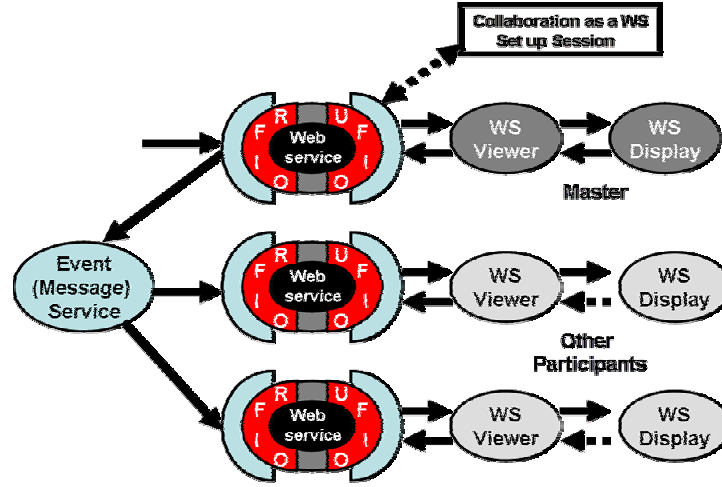


Fig. 1: Shared Web Services model using Input Ports (messages)

Project CAROUSEL at Indiana University has been investigating infrastructure for universally accessible collaboration systems. We designed and implemented “personal server” for mobile devices capable of extending “legacy” collaborative service to heterogeneous environment [20].

The collaboration in the CAROUSEL Web service is approached based on the shared input port model which we have also proposed as an object sharing model in Web service environment [1]. Each client on heterogeneous device is getting different presentation view of replica of Web service. Moreover, each user will get the non-collaborative services, such as individual scrolling or zooming the image customized based on the user profiles. The object sharing in our model of fig. 1 is achieved by intercepting the pipeline before the master Web service and directing copies of the messages on each input port of the “master” Web service to the replicated copies. To support limited capable devices, such as PDAs or smart phones, the most of elaborating processing is implemented in content server and only ready-to-use data is delivered to users. However, each user can have own object instance with non-collaborative input/output ports, and every presentation view is customized with the user profile. Therefore we expect that CAROUSEL Web service takes the advantage of shared input port model. Comparing to shared output port sharing, shared input port model offers greater customizability as each client has the freedom to accept or reject data defining the shared Web service.

3. Architecture

The CAROUSEL Web service is designed to support universal access and collaborative features by using a uniform event bus and Web service interface. We discuss about the shared input port model in next section in detail. Every message in this Web service is defined in XML and implemented with a pure Java solution.

Within this architecture, there are four major components - contest servers, aggregator, client

application and event service. To provide the rich synchronous/asynchronous collaboration features, we build message based collaboration architecture which the set of cooperating services in our framework deploy the message service for communication. Each component is linked together with their input/output ports designed to support Web services semantics.

For demonstrative purpose, we have developed a collaborative Scalable Vector Graphics (SVG) browser. SVG [12] is a 2D vector graphics standard format W3C recommended and has well structured document following XML syntax. We have adapted SVG as a format for shared export within our research of collaboration. The user-interface will display SVG image rendered by content server, provide display customization environment and process basic collaborative functionalities.

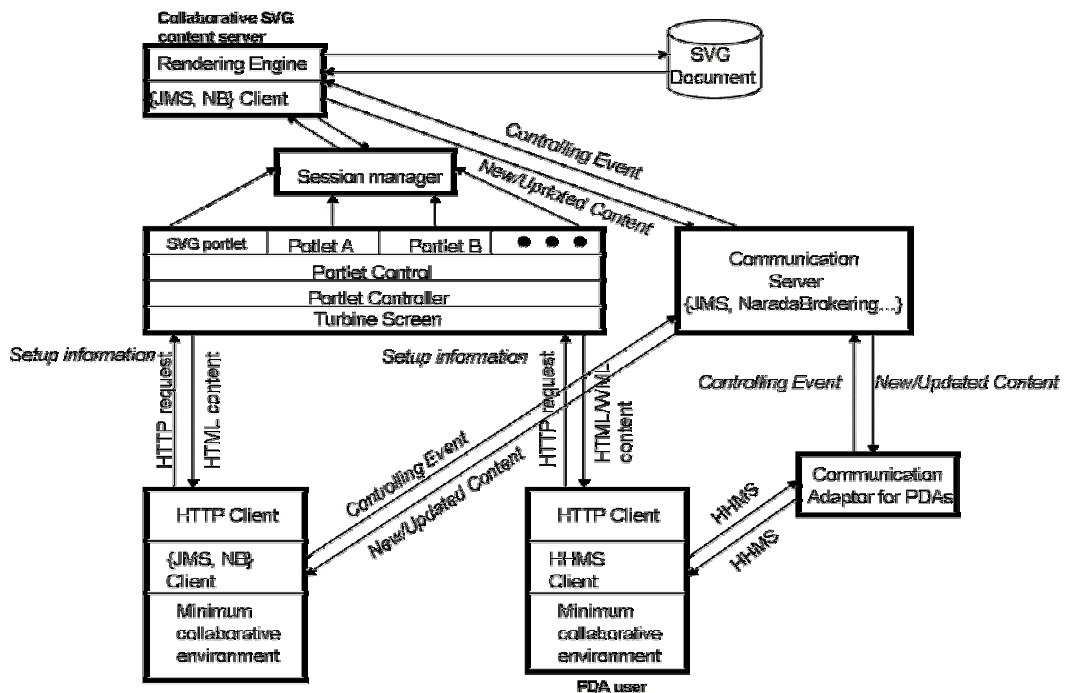


Fig. 2: The architecture of CAROUSEL Web service

3.1 Content servers

Each collaborative feature of the CAROUSEL Web service is designed as individual remote content servers in fig.2. The major requirements of content servers are,

- Content generation/processing for collaborative/non-collaborative features
- Ports facing to resources
- Customizing contents processing for pervasive users

The requests from the user to the content server can be classified as collaborative and non-collaborative requests. The bitmap based user output is manageable on user interface, while, the output rendered from filtered document should process every user command in rendering units for its quality of service. For instance, the SVG content server generates and delivers new SVG image when a user requests zoomed image only for his or her interest but not for sharing. Otherwise, every

user application must keep its own copy of original documents and the processing unit also. Every collaborative and non-collaborative event is transmitted via the NaradaBrokering messaging service. The content server is designed as a portlet of a Web services and provides input and output ports. Each portlet in the CAROUSEL Web service defines the distributed objects in a XML-based IDL (Interface Definition Language) called WSDL (Web Service Definition Language) [2]. The overall structure of every message, including input and output, are defined in WSDL. The output/input ports support dynamic communication channels to NaradaBrokering and general HTTP communication. The handoff between general HTTP communication and event service is automatically done after user's specification of its environment. If the users finish their specification on the portlet presentation view generated as HTML or WML for mobile users, the user specification is stored in user profile and the user application will get the new information about communication channels, which will be used for customized user-facing output port and collaborative input/output ports. The communication channel information includes, event service IP address/port number, topic IDs assigned for collaborative/non-collaborative events, and information about communication adaptor if the user works on mobile devices. The port to remote resources such as URLs or database is supported as well.

In order to customize the portlet presentation for each user on pervasive devices, the content server allows any number of transformations. The stylesheets specially designed for universal devices map the original document to the customized one. Every customization is performed based on the user profile from the client. This transformation not only includes stylesheets, but also processing such as the customizing of resolutions.

3.2 Event Service

CAROUSEL Web service is a message-based collaborative system. To provide messaging between the heterogeneous user network environments and Web services, NaradaBrokering [11] from the Community Grid Labs is adapted as a general event brokering system. NaradaBrokering supports the messaging models with a dynamic collection of brokers supporting a generalized publish-subscribe mechanism. NaradaBrokering supports for TCP, UDP, Multicast, SSL and raw RTP clients. For the remote resources behind of the firewall, NaradaBrokering provides the capability of the communication through firewalls and proxies. This can operate either in a client-server mode like JMS [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. As the development and administration tools, NaradaBrokering provides Performance Monitoring, and Performance Aggregation Service. We expect that the collaborative system developed based on this NaradaBrokering system provides the service to the users in heterogeneous network environments with more reliable and scalable network framework.

For mobile users, such as PDAs and smart phone, we developed a network communication adaptor for optimizing messages and providing collaborative communication environment for specific devices. The network communication adaptor provides HHMS (Handheld Message Service) [20] as a natural modification of the event service. Mobile clients have modest performance and size in comparison with traditional desktop machines. Therefore, they require particularly efficient protocols. The HHMS protocol is efficient and not very elaborate. Moreover, the communication method is very limited and specified for mobile devices also. For example, wireless Internet accessing service for smart phone or PDA phones supports only HTTP which entails request-response based transmission primitives currently. HHMS provides virtual two-way transmission primitives for collaborative network communication environment to these limited devices. Each communication service is selected from the user profile of the client's specification, and assigned to the client automatically.

3.3 Client Application

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

As an initial process, with the user specification, every user can setup their working environment for their specific machines as well as preferences in the portal presentation view. The operating systems, display types, communication method and preferred resolution are the basic factors selected in this phase.

After the initial setup, the collaboration features customized with the user profiles specified in client's setup is provided. The collaborative events such as major presenter's zooming or changing new URLs are wrapped with the collaborative application protocol and the event message is delivered to content servers via event services as an XML message. The non-collaborative events are also processed like collaborative events and the characteristics of events are identified in the tags of each event message.

3.4 Aggregator

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 separate windows and can either be distinct or partially layered on top of each other. This portlet presentation views includes all kinds of machine-dependent outputs such as bitmap display or audio streams.

4. Dataflow and Universal Access

We introduced the architecture of the collaborative application as a Web service supporting universal access in previous section. Here we discuss dataflow in the CAROUSEL Web service and how we approach the universal access and link the Web service components.

4.1 Dataflow pipeline

With regards to an object in Web services, the object is typically in some pipeline as seen in fig 3-(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 to another. Rather than a simple pipeline, one can have a complex dynamic graph linking services together [1].

We consider the output stage of this pipeline as a “document” – each with its own document object model-preferably different instances of the W3C DOM. The final user interface could be a pure audio rendering for visually challenged user, or a bitmap transmitted to a primitive client not able to perform full browser functions.

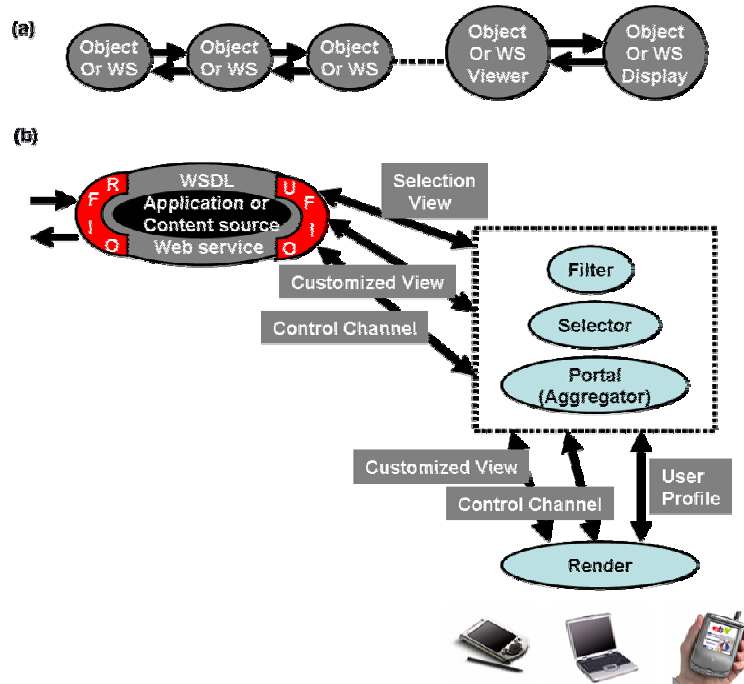


Fig. 3: (a) Web service pipeline flow from originating objects to display

(b) Architecture of Event Service and Portal to support Universal Access

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 also how the output should be modified for clients of

limited capabilities. This implies the modular pipeline of fig. 3 (a) is deficient in the sense 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 CAROUSEL Web service are designed such that it keeps original document in itself, and provides an interface to the original document to generate new output for each event. There still exist the ambiguity of reverse functionalities, but we can expect that every reverse function will get correct output with this design.

4.2 The Design of Input/Output ports

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 linked modular pipeline implemented as Web services.

Each Web service is designed with three major user-facing ports as an output port of the modular pipeline of fig.3(b). First, the main user-facing specification output ports that in general do not deliver the information defining the display but rather a menu that defines many possible views. A selector in fig.3(b) combines a user profile from the client (specified on a special profile port) with this menu to produce the “specification of actual user output” which 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.

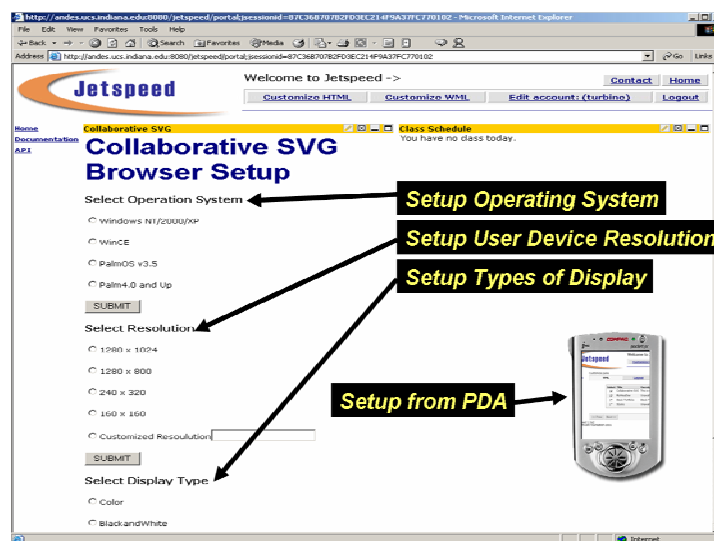


Fig. 4: User specification from client's setup

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. For collaborative SVG, specification of the output port could involve a choice of display type or resolutions in fig.4, and rendering image from SVG document transformed with CSS stylesheets [22] will be a customized user-facing output presentation in fig.5. The conversion between stylesheets could in fact involve a general filter capability of the event service as another Web filter service. It seems appropriated to consider interactions with user profiles and filters as outside the original Web service as they can be defined as interacting with the message using a general logic valid for many originating Web services.

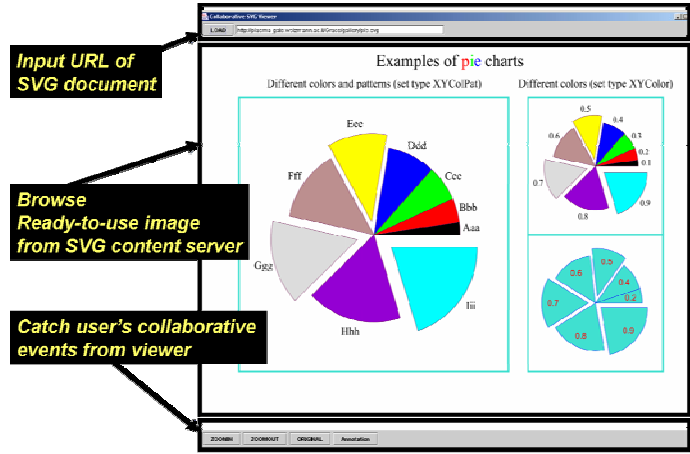


Fig. 5: User Interface with customized output display

Finally, User-facing input/output port, which is the control channel shown in fig. 3(b).

Note that in fig.3(b) 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 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. Summary and Future work

We introduced an architecture of universal accessible Web service for collaborative applications. In summary, the collaboration between users in CAROUSEL Web service is approached based on the input port sharing model for more flexible collaboration. Each Web service in this architecture provides one or more ports facing to resources and users to implement integrated collaboration

application. CAROUSEL Web service is linked to a message based event service, and every message is defined with XML. We discussed modular data flow pipeline defined in CAROUSEL Web service, and how we approach the universal access.

We will extend this work to other mobile devices such as smart phones and PDA phones using J2ME (Java 2, Micro Edition) [10]. We are also developing fully functional collaboration system as integrated Web services. It includes wrapping and integrating remote service from other vendors, such as the XML Presence Protocol (XXMP) [18] based instant messenger from Jabber Open server. The research on the communication adaptor continues in parallel. With the remarkable improvement of mobile services, there are more possibilities to adapt new communication technologies for PDAs such as JXTA and JMS style event broker. We expect that the investigation of these communication methods will provide better performance and quality 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) <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] 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>
- [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] 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/>
- [12] J. Ferraiolo, ed., *Scalable Vector Graphics (SVG) 1.0 Specification*. August 2000.
- [13] Brad A. Myers, "Using Hand-Held Devices and PCs Together.", *Communications of the ACM*. Vol. 44, Issue 11. Novembaer, 2001. pp. 34 - 41
- [14] Harakan Software, *PalmVNC*, <http://www.btinternet.com/~harakan/PalmVNC/>
- [15] Nokia, *Java VNC viewer*, <http://www.mgroeber.de/nokia.htm>
- [16] Cutting Edge Software, *Quickoffice Conference*, <http://www.cesinc.com/products>.
- [17] Dawn corp., *Feature Base*, <http://www.dawn-corp.co.jp/gis/index.html>
- [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] AT&T Laboratories, *VNC - Virtual Network Computing from AT&T Laboratories Cambridge*, <http://www.uk.research.att.com/vnc/>
- [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] Cascading Style Sheets, <http://www.w3.org/Style/CSS>

