SECOND BEST PAPER PRIZE in excellence for applied research, Presented at IEEE ComSoc sponsored
THIRD ANNUAL WIRELESS TELECOMMUNICATIONS SYMPOSIUM (WTS 2004)
May 14-15, 2004, CalPoly Pomona, Pomona, California, USA

Middleware for Multimedia Mobile Collaborative System

Xiaoyong Su, B. S. Prabhu, Chi-Cheng Chu, Rajit Gadh
UCLA - Wireless Internet for Mobile Enterprise Consortium (WINMEC)
420 Westwood Plaza, UCLA, Los Angeles, CA 90095
x.y.su, bsp, cchu, gadh@wireless.ucla.edu

Abstract

Although various Collaborative Systems have successfully improved enterprise work efficiency, Mobile Collaborative Systems, which allow collaboration via wireless network and mobile devices, still lack robust functionality and content representation support. This paper proposes a novel framework for multimedia content generation, representation and delivery, for Mobile Collaboration. A unified file format and a message queue management middleware for heterogeneous computing environment designed and employed are also discussed.

1. Introduction

In today's global market, enterprise operations are no longer limited to a central location. Tasks are usually performed at different locations around the country or even around the world; enterprise operations are connected through the use of networked computers in an Intranet or the Internet. Product design information and data can be synchronized across geographic locations. In this scenario, effective and accurate collaboration between participating team members becomes very important. In recent years, various Collaborative Systems based on Workflow Management Systems (WfMS), Groupware Systems, and Business Process Modeling Systems [1, 2] have been investigated. Some key issues for collaboration include group awareness; monitoring and control; communication and coordination within the group; data sharing and representation; and the support of heterogeneous, open computing environment. In addition the style of work and the system deployment, have implications for the design of infrastructure and applications [3]. In Collaborative Systems, several major and essential components that require study include Middleware technologies, Publish-Subscribe models, Process and Workspace Management, Community and User Management.

Recent advances in wireless communications and mobile devices have made mobile handsets, such as smart phones, PDAs and laptops, very useful and popular business tools for the executives on the move. Researchers, developers and decision makers have recognized the potential wireless networking can have on existing business processes and the opportunity to improve the services. It is anticipated that a mobile collaborative system will play a very important role in many enterprise activities [4, 5, 6, 7]. Enterprise mobility over wireless networks will need a robust infrastructure for information exchange between different business and enterprise entities. However, existing mobile collaborative systems are still not mature enough to meet this need. There are a number of issues, which make the implementation of a mobile collaborative system very challenging [8]. Firstly, on device side, the screen is small; lacks sufficient computational power (although it has been improving) and lack of 3D data graphics rendering (or even some 2D graphics) functionality; short battery life; limited storage space, etc. Secondly, on the connection side, regardless of the communication protocol used (e.g. 802.11 protocol family or wide area wireless communication protocols such as GSM, CDMA, and UMTS), wireless connections are unreliable and provide modest bandwidth. Therefore, direct migration from traditional Collaborative System to Mobile Collaborative System will be difficult and will not provide the same degree of satisfaction and sophistication to the users. The research approach in this paper addresses these issues through the framework and a prototype system demonstrates the working of the framework.

2. Review of literature and existing systems

In this section, a number of published research projects and systems were studied and the salient features of some of the related works are discussed. Dustdar and Gall [1] describe a framework for process aware distributed and mobile teamwork. A three-layer architecture that integrated workspace management, publish-subscribe, and community/user management has been implemented and tested on a peer-to-peer middleware. Roth and Unger [9] present a platform specially designed for groupware applications, 'QuickStep' running on handheld devices. Group Management, managing data and mirroring and caching have been integrated in this system. However, due to the inefficient communication infrastructure (using Bluetooth, IrDA, and Serial Connection), it is not very effective in real collaborative environment. Kirda et al. [10] have developed a service architecture for mobile teamwork. The system takes into account the different connectivity modes of users and uses XML meta-data and XML Query Language (XQL) for distributed document searches and subscriptions. Caporuscio and Inverardi [11] present a framework based on 'SIENA' [12], a Scalable Internet Event Notification Architecture developed by Software Engineering Reaserch Lab at Colorado University. In this system, event-based system and publish/subscribe mechanism are used to achieve scalability and mobility. Joseph, et al., [13] created a toolkit, 'Rover', which supports mobility transparency and mobility awareness. It has been tested in a disconnected situation by using Queued Remote Procedure Calls (QRPC) and Relocated Dynamic Objects (RDOs). Noble, et.al., [14] design an application aware adaptation system, 'Odyssey', which supports concurrent execution of diverse mobile applications and extends data distribution to video and audio. Communication bandwidth is also monitored by the system. Data delivery is based on the current available bandwidth. McKinley and Li [15] proposed a synchronous collaboration application for handheld computers called 'Pocket Pavilion'. They have employed a component based Java framework which is capable of reusing existing components or enabling users to plug-in other helper applications. The framework utilizes a 'Leader-Followers' enviroment where they collaborative session is monitored by the leader and the session is multicast to the followers using an extensible leadership protocol.

As seen in the above documented research there are a number of issues which need attention in developing a collaborative enviroment. Most existing Mobile Collaborative Systems focus on the implementation of the collaborative logic engine. The requirements for content representation in a heterogeneous computing environment are not clearly addressed. Because different platforms have different device profiles and connection capabilities, in this research it is surmised that adaptative content generation and delivery would be ideal for mobile collaborative environments. Other information representation and interaction methods such as voice input and speech/sound feedback will greatly improve the effectiveness of user interface if it can be embedded in the documents. In general, the types of media content which are commonly transmitted in a collaborative system for engineering enterprise include graphics, text messages, images, voice, and so on. These different types of content are created in different file formats and by different software components. A multimedia format that can be represented as a single, unified format and transmited as a message queue would simplify the overall communication mechanism and effectively improve the efficiency of collaboration. Developing and testing such a file format is one of the objectives of this research. So, in this paper, a multimedia mobile collaborative system based on content-aware, device-aware and connection-aware framework is presented. In the following sections, the conceptual architecture, data representation and message-oriented middleware for content generation and delivery under this framework will be discussed.

3. Conceptual Architecture of the Collaborative Environment

Content generation/delivery/presentation plays a crucial role in a collaborative system. Figure 1 shows the conceptual framework of the proposed mobile collaborative system. In this framework, there are four major components: content generation layer, communication layer, content consuming/regeneration layer and content visualization layer.

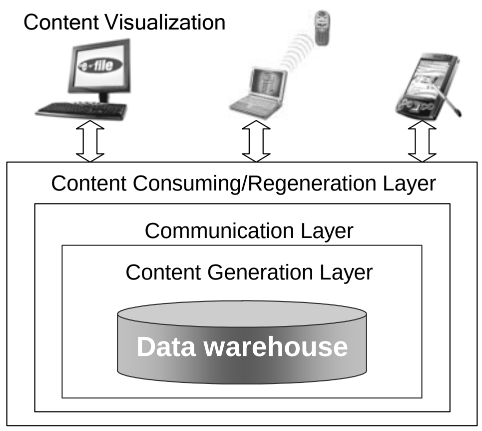


Figure 1. Framework of Mobile Collaborative Environment

Content Generation Layer: in this layer, the content server generates the unified content based on client request. Client request message consists of the device profile, previous network status, and requested URL. Content server checks the availability of the data. If the data is available for sharing, unified content (unified file format based) will be generated and delivered to the client based on the type of the requesting device and network connection.

Communication Layer: Communication layer maintains each interaction session and delivers messages between client and server. It detects the network status and decides whether to store the message queues to message buffer in case there is no network connectivity or forward them to the recipient.

Content Consuming and Regeneration Layer: In this layer, messages are assembled and sent to content visualization layer for display. If client wishes to make any modification or add annotation, content will be regenerated. The modified content can be sent to peer user or to central content server. The content regeneration is required for peer-to-peer collaboration because it allows user to interact in real-time.

Content Visualization Layer: Once the content reaches this layer, object-parsing engine parses the multimedia object to a Document Object Model (DOM) [16] structure. In this layer, a multimedia content viewer is used to display each object.

4. Functional description of the main modules of the framework

In this section the main modules of the framework are discussed in details, explaining the functionality of the different features.

4.1. Data Representation

A Unified Media Description Language (UMDL) is proposed in the architecture. UMDL data representation adopts the extendable features of Extensible Markup Language (XML), which has been widely used in the industry, to embed multimedia information into a single, unified file format (UFF). In UMDL, several basic objects are defined: text, voice, image and graphics. These objects are organized in a DOM structure. DOM structure allows the representation of a document as a hierarchical tree of 'node' objects, where each of the nodes is defined syntactically and semantically a priori. This representation allows contextual parsing of the document. In the current context, graphics is defined using vector representation, which can be described by plain text syntax. There are two methods to present voice information. The first method is to present binary format of voice data into XML metadata format. The second method is to convert voice data into text by using voice recognition. Image is always represented in metadata format. Figure 2 shows a sample of UMDL. It uses XML syntax to describe objects such as voice, image, 3D geometry, annotation, etc. Figure 3 shows the working model of unified file format (UFF, a multimedia file format based on UMDL). In this model, multimedia data is organized and transmitted by UFF format. Both client and server have the capability to parse and generate UFF. UFF can be used to organize content hierarchically by using links.

```
<?xml version="1.0" encoding="UTF-8"?>

<xml>
  <voice encode = “base64”>pcEATSAJBAAA8BK/AAAAAAAABAAAXwoAAA4AYmpiau</audio>
  <voice encode = “TextSpeech”> welcome to use mobile collaborative system </audio>

  <image encode =”base64”>
          <data>“/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwc”</data>
  </image>

  <graphics>
          <3dgeometry>
                    <facemesh name = “face1” ………>
                    </facemesh>
          </3dgeometry>
  </graphics>

   <text value = “test”></text>
</xml>
```

Figure 2. Unified Media Description Language

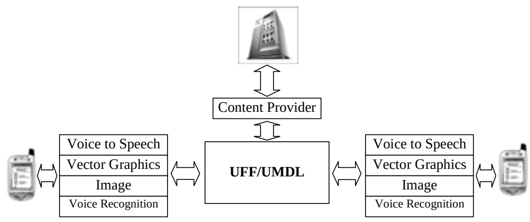


Figure 3. Unified file format working model

4.2. Content Generation, Delivery and Communication Layer

In this layer, a message-oriented middleware (MOM) based content generation/delivery service that handles the content generation and delivery is proposed. This service is called Smart Content Service (SCS). Figure 4 shows the components and relationship among the components.

In SCS, content provider or content consumer generates content. Multimedia information is separated and embedded into UFF and sent to the message agent. Message agent negotiates with content service to finish the content delivery, forward or buffering. Content service provides a way for applications to communicate with each other. Session control management is used to maintain the status of an application. A profiles database that keeps the hardware configuration of the mobile device platform is also maintained. Each time when a client sends a request to the content service, the platform of the mobile device is identified and the server will generate the application content based on the specific profile. Content operations (creations/modifications) are executed on content provider (server side) and content consumer (client side). These components are explained in detail in the following sections.

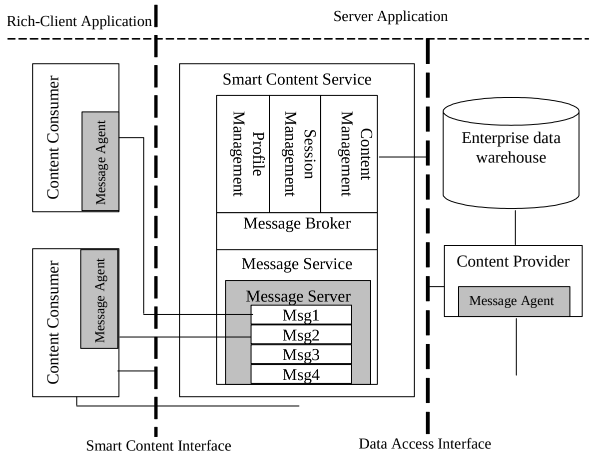


Figure 4. Architecture of Smart Content Service

4.2.1. Message Service

Message service uses message queuing to manage and dispatch messages. It consists of message agent and message server. For the convenience of discussion on message service, some terminologies need to be defined [17]:

- **Producer**: the object that generates messages.
- **Consumer**: the object that receives messages.
- **Message**: a content entity of information. It can be an XML expression, a text string, an intelligent agent or anything that both producer and consumer can understand.
- **Message Agent**: Message agent has two major duties. Firstly, it handles message queuing, "store and forward" messages. When the connection is not available, it stores messages. Once the connection is restored, message agent sends the stored messages to the message server, then message server dispatches it to objects based on the destination of the messages. Secondly, message agent translates messages, and provides the usable information to upper layer applications.

In the current model, message agent can be both producer and consumer. When it sends information to other application, it acts as a producer; when it receives a message from message server, it acts as a consumer.

The message entity includes destination, content type, priority, time-to-live and message body. Figure 5 shows the format of a message.

Destination	Type	Priority	Time-To-Live	Content
Content				

Figure 5. Message Format

There are two methods for dispatching the messages. 1) Message server delivers messages based on the destination. If the destination cannot be accessed, the messages stay in the queue for next attempt until the Time-To-Live expires. 2) Client application sends a message request to message server, if the message server has messages for this client application, messages are delivered to the client application. A message is considered consumed when a client application receives a message and sends a feedback to the server. The server will delete consumed messages from the message queue automatically.

4.2.2. Message Broker

Message Broker acts as a middleman, which decodes a message to upper layer objects or encodes the data to be transmitted by the message service. There are three types of messages dealing with the three upper layer objects - profile, session, and content. Message broker receives a message from message service, decodes the message, and then delivers it to the corresponding objects. When the upper layer objects want to send a message to client application, they notify message broker the destination and content. Message broker encapsulates the information in a message, and sends it to message server.

4.2.3. Session management Object

The session management module is shown in Figure 6. The session management module keeps the authentication and session persistence. A session is established each time a client application logs in. Each session has a unique tag. The session records each transaction of an application. User can stop, suspend, and resume a session at anytime.

A session can maintain the progress of an application. Once a mobile user (client) loses the connection with smart content provider, the session will suspend until its time-to-live expires. When the mobile user reconnects to smart content provider, she/he can resume his work based on the transactions. The session management decides which transaction will be re-executed to guarantee there is no duplicate execution and to maintain data consistency. It is also desirable to keep the transactions independent of the client platforms to facilitate the client to reconnect from a different platform/device.

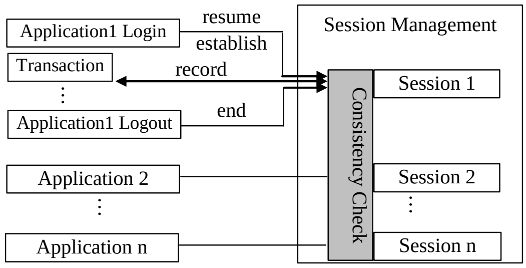


Figure 6. Session Management Architecture

4.2.4. Profile Management Object

Profile management object manages the profile of mobile users. A mobile user profile has the following information:

- Hardware resource: hardware resource includes the CPU frequency, usable memory, screen size, and hardware architecture.
- Operating system: operating system information includes the type and the version of OS.
- Connection status: Connection status includes communication protocol, bandwidth, and network status (signal strength)

The principle of profile management is the same as that of session management object. Profile management object requests the client profile when mobile user logs in. This object maintains the relationship of each profile with the respective session or sessions. The system uses this profile to determine what kind of content can be delivered to the requesting client.

4.2.5. Content Management Object

Content management object interacts with profile management object and session management object. It is a dynamic content provider. Content management object provides content based on three factors: 1) client request; 2) client profile; 3) session. Once the profile of a session changes (this indicates that user has switched to another mobile device), the content management object will generate the respective content for that mobile device. For example, if a mobile user uses a laptop to access the current collaborative system, the system determines that rich content can be displayed on the client application properly, so picture, audio, video and other large multimedia content are sent to client-application. On the other hand, if a mobile user uses a PDA to access the system, the system will be aware that it should deliver the smaller size files or text message to that client.

5. Prototype of Multimedia Mobile Collaborative System

5.1. Architecture of prototype

Based on the framework mentioned above, a prototype of Multimedia Mobile Collaborative System being implemented as shown in figure 7.

In this mobile collaborative approach, a mobile user can communicate with group members by using email, chat and voice message, share documents and access enterprise database, etc. The basic modules of the system include process monitoring, work assignment, member management, documents sharing/representation and communication. Process monitoring is a module that a group leader can use to monitor progress of a project or design and take necessary actions. Group leader uses work assignment and member management functions to manage members, create schedule and assign duty to members. There are several methods for communication, which include chat, voice message and email.

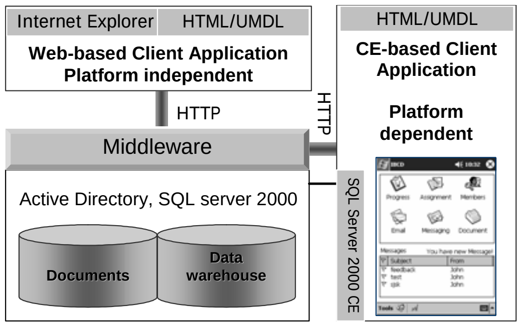


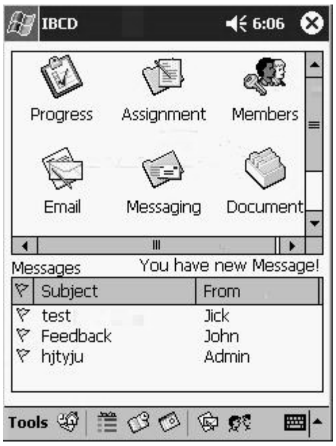
Figure 7. Architecture of a mobile collaboration system

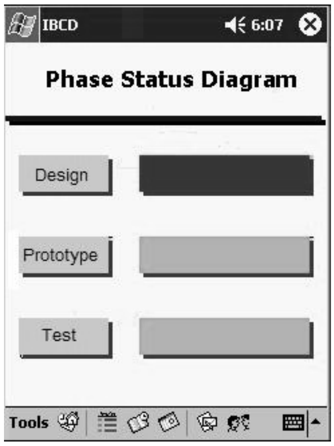
The system is a client-server application. Server side consists of SQL Server 2000 Database, SQL Server 2000 CE Agent and middleware, which will provide content to the client. Client offers read, write, data manipulation and communication functions through agents.

5.2. Screen shorts

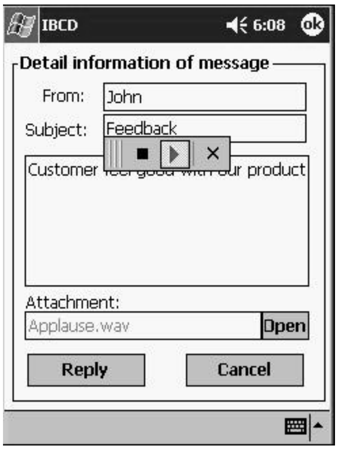
The following figures show the screen shots of the various steps in a typical collaboration session. The screens show the stages such as the main interface, the status of the collaborative activity, the hierarchical representation of the content, sample engineering content and user interaction, etc. The current version of the client is ported to Pocket PC (Windows CE 3.0) on iPAQ h5450. The device was connected to the net over 802.11b.

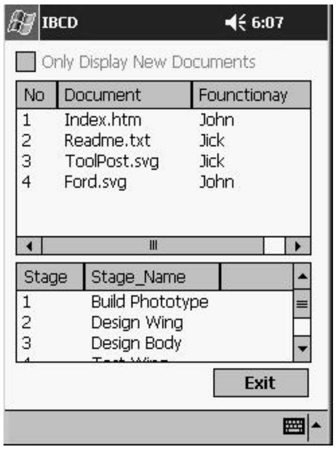
- Main interface and progress



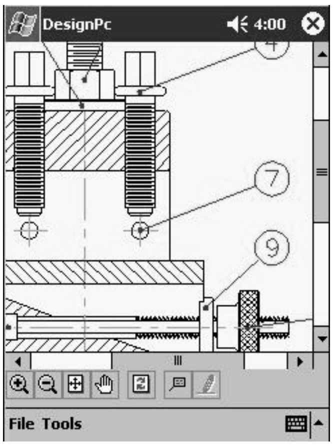


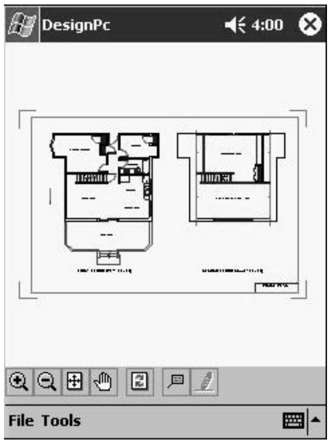
- Message and UFF based multimedia content organization





- Content visualization on client device

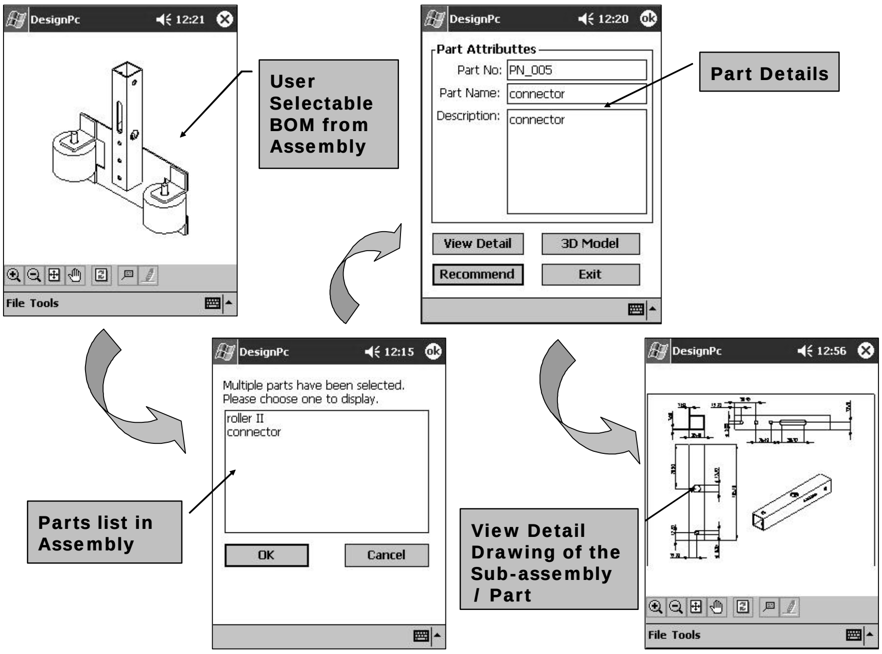




Part Drawing

House Floor Plan

- Hierarchical content organization by using UFF.



Accessing product assembly, bill-of-materials, right up to individual part drawings

6. Conclusion and future work

A consolidated data representation scheme for multimedia content – 2D/3D graphics, audio/voice and video with contextual processing capability of the content enabling optimum data transmission proposed in this work along with the ability of the middleware framework to support device, user and session management is anticipated to be a significant contribution for mobile collaboration using heterogeneous small devices such as PDAs and Laptops.

With the current design of the framework and the Unified Media Descript Language, it has been possible in the prototype system to effectively represent multimedia information into a single data file. It was observed that this representation simplified the data stream between client and server. The performance of the system was found to be satisfactory in a heterogeneous computing environment containing PDAs and laptops. It was observed that the data representation scheme permitted content exchange in a collaborative design system. Currently, only text, image and graphics content are supported by UFF. The capabilities of the framework are being extended to support other multimedia content including audio, voice and video.

This proposed middleware architecture can be applied to mobile collaboration activities. It is especially suitable for field services such as repair, maintenance. In this situation, it requires that people use mobile device with wireless wide area network connection to access engineering data. Original engineering data on the enterprise server has to be adapted and converted into UFF then send to mobile device. Another potential usage of this architecture is content exchange in a mobile peer-to-peer environment. Individual mobile client can use this framework to share multimedia content without having a server. Service agents can provide content service and messaging service for peer client.

This research demonstrates that mobile collaborative environments are feasible and with further development in devices and the network technologies it will become a potent tool for forward looking enterprises in their quest for productivity improvements.

Acknowledgment

We acknowledge sponsorship from Intel and UCLA-WINMEC (http://winmec.ucla.edu) in the support of this project.

Reference

[1] Schahram Dustdar, Harald Gall. "Architectural Concerns in Distributed and Mobile Collaborative systems", 11th Euromicro Conference on Parallel, Distributed and Network-Based Processing, Feb., 05-07, 2003, Genoa, Italy

[2] Henrique J. Domingos, at el. "A workflow Architecture to Manage Mobile Collaborative Work", Proc. of EPCM '99, Nov., 13-14, 1999, Tomar, Portugal.

[3] Edwards Keith, Mynatt Elizabeth, Peterson Karin, "Designing and Implementing Asynchronous Collaborative Applications with Bayou", Proc. of 10th ACM Symposium on User Interface Software and Technology (UIST '97), Banff, Alberta, Canada, Oct. 14-17, 1997.

[4] Brad A. Myers. "Using hand-Held Devices and PCs Together". Communications of the ACM, vol.44, Issue 11. November, 2001, pp 34-41.

[5] Geoffery Fox, at el, "Integration of Hand-Held Devices into Collaborative Environments", In Proceedings of 1st International Workshop on Wired/Wireless Internet Communications (WWIC2002) Conference, Jun., 24-25, 2002, Las Vegas, USA.

[6] Jianzhong Mo, Chi-Cheng Chu, B. S. Prabhu, Rajit Gadh, "On the Creation of a Unified Modeling Language based Collaborative Virtual Assembly / Disassembly System", 8th Design for Manufacturing Conf., ASME 2003 Intl. Design Engg., Technical Conferences and the Computers and Information in Engineering Conference (DETC2003/DFM-48160), Sep, 2-6, 2003, Chicago, USA.

[7] Shyamsundar N and Rajit Gadh, "Internet-based Collaborative Product Design with Assembly Features and Virtual Design Spaces", Computer-aided Design, Vol. 33, No. 9, 2001, pp 637-651.

[8] Xiaoyong Su, B. S. Prabhu, Rajit Gadh, "Mobile Graphics on PDA : Introduction to SVG", Tech. Report : UCLA-WINMEC-2003-101, Wireless Media Lab, The Henry Sameuli School of Engg. and Applied Science, UCLA.

[9] Jorg Roth, Clause Unger, "Using Handheld Devices in Synchronous Collaborative Scenarios", Personal and Ubiquitous Computing, Vol. 5, No. 4, London, Dec. 2001, 243-252

[10] Engin Kirda, et al. "A Service Architecture for Mobile Teamwork", Intl. Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 4 (2003) 447-467.

[11] Mauro Caporuscio and Paola Inverardi, "Yet Another Framework for support Mobile and Collaborative Work", Intl. Workshop on Distributed and Mobile Collaboration (DMC2003). IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-2003). Linz, Austria. June, 2003.

[12] Software Engineering Reaserch Lab, Colorado University. "SIENA (Scalable Internet Event Notification Architectures)" [http://www.cs.colorado.edu/users/carzanig/siena/]

[13] Joseph A. D., Tauber J. A., Kasshoek M. F., "Mobile Computing with Rover Toolkit", IEEE Transactions on Computers, Vol. 46, No.3 March 1997 337-352.

[14] Noble B. A., Satyanaryanan M., Narayanan D., Tilton J.E., Flinn J., Walker K., "Agile Application-Aware Adaptation for Mobility", Proceedings of the 16th ACM Symposium on operating System Principles, Oct. 1997, St. Malo, France.

[15] Philip K. McKinley and Ji Li, "Pocket Pavilion : A Synchronous Collaborative Browsing Application for Wireless Handheld Computers", Tech. Report – MSU-CPS-99-38, Dept. of Computer Science and Engg., Michigan State University, Michigan, Nov. 1999.

[16] Document Object Model, http://www.w3.org/DOM/

[17] Silvano Maffeis, "Middleware Support for Application-to-Application Wireless Messaging", Whitepaper from Softwired, Bonstetten, Switzerland [http://www.softwired-inc.com/people/maffeis/