Middleware for Multimedia Mobile Collaborative System

Xiaoyong Su, B. S. Prabhu, Chi-Cheng Chu, Rajit Gadh
Wireless Internet for the Mobile Enterprise Consortium (WINMEC) - wireless@winmec.ucla.edu,
420 Westwood Plaza, University of California, Los Angeles, CA 90095.

Abstract

Although various Collaborative Systems have successfully improved enterprise work efficiency, Mobile Collaborative Systems, which allows collaboration via wireless network and mobile devices, still lack functionality and content representation supporting. This paper proposes an infrastructure, which focuses on Multimedia Content Generation, Delivery, and Representation, for Mobile Collaborative System. A unified file format and a message queue management middleware for heterogeneous computing environment will also be 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 tightly connected through the use of networked computers such as an Intranet or the Intranet. Product design information and data can be synchronized across geographic location. 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] 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 Collaborative Systems, several major and essential components that require study include Middleware, Publish-Subscribe, Process and Workspace Management, Community and User Management.

Recent advances in wireless communication and mobile devices have made mobile handsets, such as PDAs, very useful and popular business tools while traveling. Various IT and business researchers have recognized the potential wireless networking has to change the world's business model. Enterprise mobility over wireless network needs massive information exchange between different business units. Thus, a highly mobile collaborative system plays a very important role in enterprise operation. However, existed mobile device based Collaborative Systems are still not mature enough to meet this need. Implementing Collaborative System on mobile devices, especially on PDAs, faces more challenges than implementing Collaborative Systems in a wired, desktop environment. Firstly, on device side, the screen is small; computational capability is weak (although it is been improved rapidly) and lack of 3D (or even some 2D) 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 Public Wireless Communication Protocols like

CDMA, GSM, 3G), wireless connections are unreliable and have low bandwidth. Therefore, direct migration from traditional Collaborative System to Mobile Collaborative System simply won't work.

In previous researches, Schahram Dustdar and Harald Gall [1] describe a framework for process aware distributed and mobile teamwork. A three-layer architecture that integrated workspace management, Publish-Subscribe, and Community and User Management has been implemented and tested on a Peer-to-Peer middleware. Jörg Roth and Claus Unger present a platform specially designed for groupware applications, QuickStep [2] 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. Engin Kirda et al. develop a service architecture for mobile teamwork [9]. They take account of the different connectivity modes of users and use XML meta-data and XML Query Language (XQL) for distributed document searches and subscriptions. Mauro Caporuscio and Paola Inverardi [10] present a framework based on SIENA [11], A Scalable Internet Event Notification Architectures developed by Software Engineering Reaserch Lab at Colorado University. In this system, event-based system and publish/subscribe mechanism are used to acheve scalability and mobility. Joseph A. D. et al. created a toolkit, Rover [4], which support mobility transparent and mobility aware. It has been tested in a disconnected situation by using Queued Remote Procedure Calls (QRPC) and Relocated Dynamic Objects (RDOs). Brian D. Noble designs a application aware adaptation system, Odyssey [5], which supports concurrent execution of diverse mobile applications and extents data distribution to video and audio. Communication bandwidth are monitored in the system. Data dilevery is based on the current available bandwidth.

Among the above Mobile Collaborative System, most of them focus on Collaborative Logic Implementation. The requirements for content representation on a heterogenous computing evironment are not clearly defined. Because different platforms have different device profile and connection capability, adaptative content generation and delivery should be used for data sharing. Other information representation motheds such as voice input and speech/sound feedback will greatly improve the effectivness of user interface if it can be embedded in the documents. In generally, the types of media which are commonly transmitted in a Collaborative System include graphics, text messages, images, voice, and so on. These information have been created in different file format and have been handled by different conponents. Multimedia format that is represented in a single, unified format and transmited as a messages queue would simplify the overall communication mechanism and effectively improve the efficiency of collaboration. In this paper, a content-awared, device-awared and connection-awared framework for Multimedia Mobile Collaborative System is presented. In the following sections, Conceptual Architecture, Data Representation and Message-Oriented Middleware for content generation and delivery under this framework will be discussed.

2. Conceptual Architecture in Collaborative Environment

The primary idea of a collaborative system is that members in a group can share data, exchange idea and access enterprise data warehouse, etc. Content Generation/Delivery/Presentation plays a crucial role in a collaborative system. Figure 1 shows the conceptual framework of proposed mobile collaborative system. In this framework, there are four major components: Content Generation Layer, Communicate Layer, Content Consuming / Regeneration Layer and Content Representation layer.

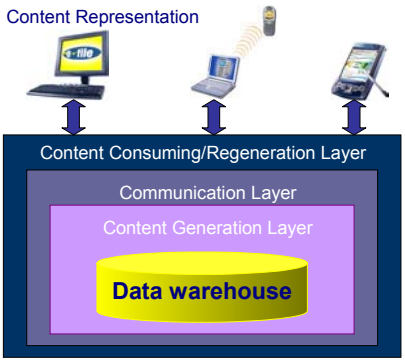

Figure 1. Topology of a mobile collaborative environment

Content Generation Layer: in this layer, Content Server generates the unified content based on client request. Client request message consists of 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 deliver to the client based on the type of the requesting device and network connection.

Communication Layer: Communication Layer maintains each conversation session and delivers messages between client and server. It detects the network status and decides whether store the message queues to message buffer or forward them to recipient.

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

Content Representation Layer: Once the content reaches this layer, object-parsing engine parse the multimedia object to a Document Object Model (DOM) [12] structure (will be discussed later). Is this layer, a multimedia content viewer is used to display each Object.

3. Data Representation

A Unified Media Descript Language (UMDL) will be used in the current architecture. UMDL data representation adopts the extendable features of Extendable Markup Language, which have been widely used in the industry, to embed multimedia information into a single, unified file format (UFF). In UDML, several basic objects are defined: text, voice, image and graphics. These object organized in a DOM structure. In the current context, graphics are defined as vectorized graphics 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 UDML. It's use XML syntax to describe objects. Figure 3 shows the working model of unified file format (UFF, a multimedia file format based on UDML). In this model, multimedia data organized and transmit by UFF format. Both client and server have the capability to parse and generate the UFF.

```
<?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>
       <3dgeometrics>
             <facemesh name = “face1” ……>
             </facemesh>
       </3dgeometrics>
 </graphics>

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

Fig. 2 Unified Media Descript Languange

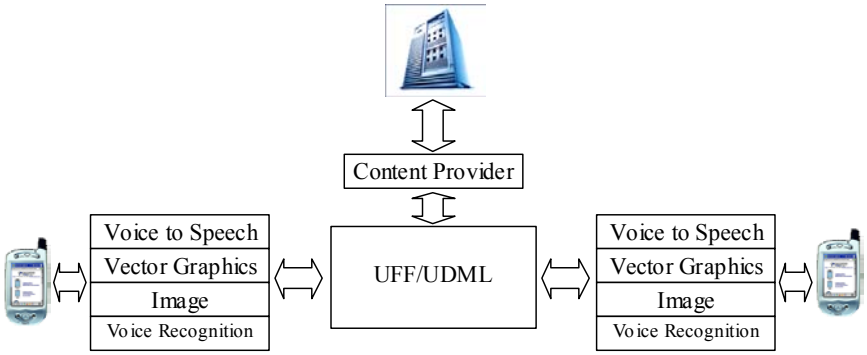

Fig. 3 Unified file format working model

4. Content Generate/Delivery/Communication Layer

In the current research, a Message Oriented Middleware (MOM) based content generation/delivery service that handles the content generation and delivery is proposed. Figure 4 shows the components and the relationship between components.

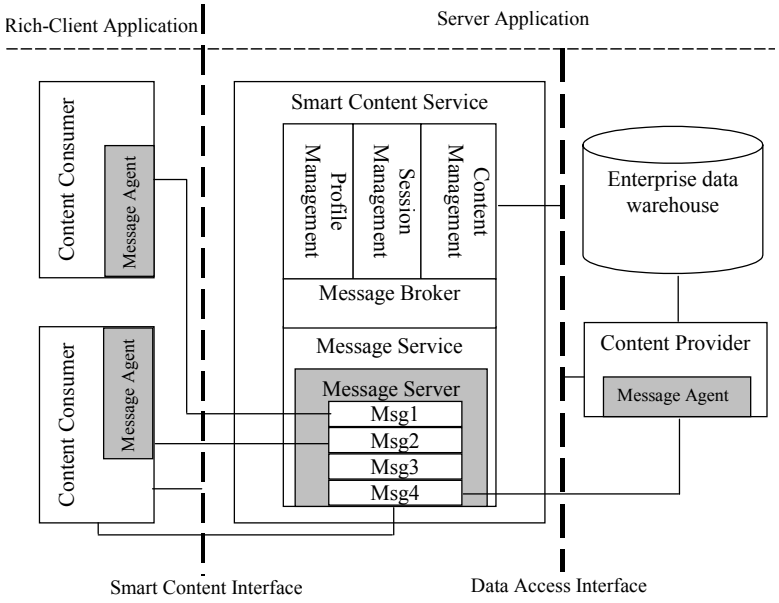

Figure 4. Architecture of Smart Content Service

In this architecture, contents are generated by Content Provider or Content Consumer. Multimedia information are separated and embedded into UFF and sent to 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 keep the setting of an application and its status. 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 server will generate the application content based on different profiles. Content operations (creations/modifications) are executed on Content Provider (Server Side) and Content Consumer (Server Side). These components are explained in detail in the following sections.

4.1 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 [3]:

- Producer: the object that generates messages.
- Consumer: the object that receive 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 message server, then message

server dispatch 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 receive message from message server, it acts as a consumer.

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

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 access, the messages stay in the queue until next attempt. 2) Client application send 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 received a message and sent a feedback to the server. The server will delete consumed messages form message queue automatically.

4.2 Message Broker

Message Broker acts as a middleman, which translates a message to an upper layer objects and translates the data to message. Based on current application model, there are three types of message, 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.

4.3 Session management Object

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

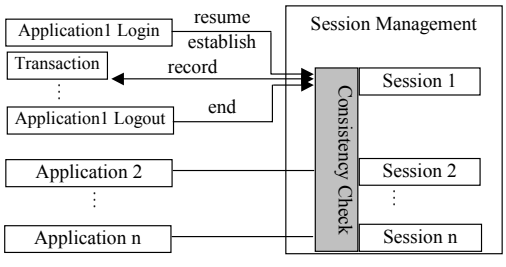

Figure 6. Session Management Architecture

A session can maintain the progress of an application. Once a mobile user (client) lost the connection with Smart

Content Provider, the session will suspend until its time-to-live has expired. 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 no duplicated execution to keep data consistency. It is also desirable keep the transactions independent of platforms in case the client reconnects from a different platform/device.

4.4 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 (perfect or poor)

The principal of Profile Management is the same as that of Session Management Object. Profile Management Object requests the client profile when mobile user login. Each profile will be put on the same tag of respective session. The current system uses this profile to determine what kind of content will be delivered to client-application.

4.5 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's request; 2) Client profile; 3) Session. Once the profile of a session changes (this indicates that user switch to another mobile device), the Content Management Object will generate the respective content for that mobile device. For example, if a mobile user uses laptop to access the current Collaborative System, the system determines that a rich content can be display on 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 PDA to access the system, the system will aware that it should deliver the smaller size files or text message to that client.

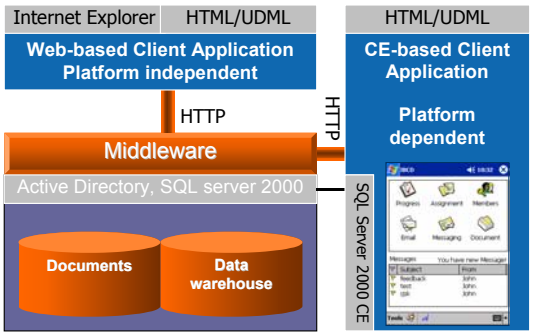

Figure 7 Architecture of a mobile collaboration system

5. Prototype of Multimedia Mobile Collaborative System

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

In this mobile collaborative system, 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 including 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 make some actions. Group leader uses work assignment and member management to manage members, creates schedule and assigns duty to members. There are several methods for communication, which include chat, voice message and email.

Two different types of applications will be implemented in the current system. One is the mobile device based application, which runs on PocketPC; the other one is web-based application, which is platform independent. Both of them are Client-Server application. Server side consists of SQL Server 2000 Database, SQL Server 2000 CE Agent and middleware, which provide content to client. Client offers read, write, data manipulation and communication functions through agents.

6. Conclusion and future work

Unified Multimedia Descript Language can effectively represent multimedia information into a single data file, which simplified the data stream between client and server. It is useful in a heterogeneous computing environment and simplifies the data representation and sharing of a collaborative design system. Currently, only text, image and graphics has been integrated into UFF, further other multimedia content including audio, voice and video will be integrated into UFF.

While industry is trying to solve the connectivity problems at hardware side; wireless connection remains unstable. The proposed Content Service is still on the way of implementation. We believe, once it is put into practice, it can be served as an effective way to solve the Collaboration problems that we are facing right now.

Reference:

[1]. Schahram Duster, Harald Gall. "Architectural Concerns in Distributed and Mobile Collaborative systems", http://portal.acm.org/citation.cfm?id=568852&coll=portal&dl=ACM&ret=1#Fulltext

[2]. Jorg Roth, Clause Unger, "Using Handheld Devices in Synchronous Collaborative Scenarios, http://citeseer.nj.nec.com/roth00using.html

[3] Silvano Maffeis, "Middleware Support for Application-to-Application Wireless Messaging", http://www.softwired-inc.com/people/maffeis/publications.html

[4] 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.

[5] 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

[6] Henrique J. Domingos, at el. "A workflow Architecture to Manage Mobile Collaborative Work". http://gsd.di.uminho.pt/epcm99/pdf/sessao3/workflow.pdf

[7] Geoffery Fox, at el, "Integration of Hand-Held Devices into Collaborative Environments". http://grids.ucs.indiana.edu/ptliupages/publications/PDA_IC2002.pdf

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

[9] Engin Kirda, et al. "A Service Architecture for Mobile Teamwork", http://portal.acm.org/citation.cfm?id=568850&coll=GUIDE&dl=GUIDE&ret=1#Fulltext

[10] Mauro Caporuscio and Paola Inverardi, "Yet Another Framework for support Mobile and Collaborative Work". http://www.di.univaq.it/~caporusc/publications/ci_dmc03.pdf

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

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