Optimized Communication using the SOAP Infoset For Mobile Multimedia Collaboration Applications

Sangyoon Oh[1,2], Hasan Bulut[1,2], Ahmet Uyar[1,3], Wenjun Wu[1], Geoffrey Fox[1,2]

[1] *Community Grids Computing Laboratory, Indiana University, USA*
[2] *Computer Science Department, Indiana University, USA*
[3] *Department of Electrical Engineering and Computer Science, Syracuse University, USA*
{ohsangy, hbulut, wewu, gcf}@indiana.edu, auyar@syr.edu
Indiana University Research Park, 501 N Morton St. 222, Bloomington, IN, 47408, USA
Fax: 1-812-856-7972

ABSTRACT

Mobile computing is growing in popularity due to continuing improvements in mobile devices and their connectivity. Collaboration systems enable users at geographically distributed locations to collaborate each other. Community Grids Lab's GlobalMMCS is a Web Services based, integrated videoconferencing system that allows heterogeneous clients to join the same session. Integration of mobile devices into the Web Services based collaboration system will give mobility to users and heterogeneity to the system. However, there are performance limitations if we use the current SOAP approaches to integrate mobile applications with Web Services based collaboration systems, especially for multimedia applications. In this paper, we propose a SOAP Infoset preserving Flexible Representation and its specialization for mobile computing environment – Handheld Flexible Representation (HHFR). We provide a flexible representation of binary encoding with a description file for message format and include a QoS framework in the architecture, such as reliability and security.

KEYWORDS

Mobile Computing, SOAP Infoset, Performance, XGSP, Audio/Video Conferencing

1. INTRODUCTION

Mobile computing and Web Services are becoming popular in collaboration systems, with mobile computing adding heterogeneity to collaboration systems and Web Services providing interoperability. Recently, information access from mobile phones has become easier with the widespread availability of packet-switched, always-on networks. As cellular phones are getting "smarter" and PDAs are becoming more "connected", more and more mobile devices are sharing resources with other distributed systems. Web Services technology helps us to integrate independently developed collaboration system components on various platforms in an interoperable distributed system. By using open standards – SOAP and WSDL, process invocations and data exchanges between software applications can be written in different programming languages.

However, the current SOAP approaches to Web Services possess performance limitations in integrating mobile applications with Web Services-based collaboration systems, especially for multimedia applications. The inevitable mobile computing characteristics – narrow bandwidth, limited computation, and small memory space – make processing SOAP messages (parsing, validation, and transformation) consume valuable resources [1]. In addition, multimedia applications need and consume substantial resources because of their high rate data exchange and media processing. High performance SOAP encoding is an open research area [2], [3], [4], and Web Services in mobile environment also need to overcome such performance limitations.

We propose a SOAP Infoset [5] preserving *Flexible Representation* and its specialization for mobile computing environment – *Handheld Flexible Representation (HHFR)*. Our approach faithfully preserves the SOAP semantics with a binary data format for high performance message transmission and processing. DFDL [6] allows us to generate any representation from binary data format specified by the DFDL description language and processed by the associated library. Alternatively conventional SOAP/XML encoding is one of message representation option to support conventional Web Services. In addition to performance issues, HHFR includes a QoS framework in the architecture to support reliability and security.

This paper describes a background of our project, our proposed flexible representation in general terms, and gives an overall architecture for our specific Handheld Flexible Representation that is suitable for mobile application with a stream of messages. Later, we give an example of an Audio/Video mobile collaboration application in Flexible Representation of Web Services. Conclusion and future works are in the final section.

2. BACKGROUND

There are several important projects from academia and industry that aim to overcome performance limitations of the SOAP encoding. Extreme! Lab at Indiana University researched SOAP negotiation and binary data transfers [2], [3] for the scientific computing field. Large data sets including arrays are common in scientific computing, which is one of the most important areas of grid computing. Thus, the condition they are facing with the conventional Web Services is similar to the constraint of mobile computing. Both need to overcome performance limitations of SOAP.

The W3C Workshop on Binary Interchange of XML Information Item Sets (Infoset) [7] lists dozens of position papers from various institutes [4], [8], [9]. The purpose of the workshop is to study methods to compress XML documents. Sun's Fast Web Services [4] uses a binary encoding for the SOAP payload. The higher level protocols (WSDL for contract definition of service etc.) remain unchanged, thus you could use standard SOAP-XML for development, and have a switch that turns on the binary protocol for production deployment. DFDL [6] is a descriptive language that is proposed to describe a file or content in a binary format for Grid computing. The language is being designed to be processable via standardized parsers that read a DFDL description, along with a file or a stream of binary data, to produce structured output - XML. Description files in DFDL and associated library provide us to encode/decode binary data into/from XML or any preferred representation.

Handheld Message Service (HHMS) [10] is a general mobile communication framework that we developed to help mobile application developers. HHMS provides a core subset API of JMS for mobile application, allowing them to be seamlessly connected to a conventional publish/subscribe system using a small foot-print user library and a server-side gateway. An advantage of using publish/subscribe system in wireless environment is studied in many projects [11], [12]. It decouples message sender and receiver by delivering messages to a topic not a static address. Message queuing in conjunction with the publish/subscribe paradigm of HHMS provides a reliable message delivery mechanism. HHFR is a part of HHMS programming model that provides a base communication protocol for mobile devices. The session chooses its transport protocol at the beginning, and the stream of messages is transmitted over the chosen transport. Currently our HHMS implementation provides two transport protocol choices - TCP and HTTP.

Oracle Web Conferencing is a real-time web conferencing system offered by Oracle Real Time Collaboration (RTC) software [13]. It provides co-browsing between collaborators, whiteboard usage, polling, chatting, and voice streaming. It is a good example of current collaboration systems movement towards Web Services. By integrating disparate and independently developed applications using Web Services, collaboration systems will have rich features while avoiding redundant developments. RTC Integration Services uses Web Services paradigm of sending XML messages to facilitate easy integration between existing business applications.

We have developed XML based General Session Protocol (XGSP) [14], [15], a Web Services based conference control framework, to integrate various videoconferencing systems such as H.323-based systems, AccessGrid [16], and SIP [17] -based systems. Any MBONE tool can join a XGSP session to send/receive A/V stream. In addition to real-time videoconferencing clients, XGSP also supports streaming media clients so that those streaming media clients can receive real-time videoconferencing streams. RealPlayers on cellular phones can receive streams in videoconferencing sessions in RealMedia format.

3. GENERAL FLEXIBLE REPRESENTATION

A data model and its representation is a key factor for interchanging complex data format in distributed application developments. There is “no silver bullet” representation for every application domain. Thus, in application developments, it is important to have a good balance between data transparency for interoperability and data format efficiency for the performance. For instance, XML is a human-readable, ubiquitous, and self-descriptive form of the data representation. SOAP provides the message architecture to allow independently developed disparate software modules (components) interact seamlessly. It, however, imposes serious performance overheads such as data conversions from/to textual format and parsing the structure, particularly in applications, such as A/V conferencing, mobile applications, and high-performance parallel computing. It is hard to address this problem on the individual message level, but it appears to be possible to combine SOAP structure and binary message format in high performance stream processing.

3.1 Flexible Representation of Data

Our Flexible Representation for a SOAP Infoset defines a collection of schemes that includes a binary encoding scheme of SOAP message, a reliable messaging scheme, a security model and a negotiation specification. It demonstrates an efficient and reliable way of transmitting messages in Web Services. The Handheld Flexible Representation (HHFR) is a specialization of this for mobile computing environment. Representations are transformed by filters. Filter F_{12} transforms Representation A_1 to A_2. For instance, a white board application in Web Services needs a pair of points to draw a line while if you draw a wire-line trace, you might need a sequence of point data. An example XML representation of wire-line trace data for a white board application, which we name it as representation A_1, can be as follows:

```
<wb:wireline>
    <wb:points>
       <wb:x>0</wb:x>
       <wb:y>0</wb:y>
    </wb:points>
    .....
    <wb:points>
       <wb:x>100</wb:x>
       <wb:y>100</wb:y>
    </wb:points>
</wb:wireline>
```

A binary format representation with tokens could be &$0$0......$1100100$1100100& – a representation A_2 (after removing some zeros). A description file of the filter F_{12} transforms the representation A_1 to A_2 using some well defined scheme. For example, a description file in DFDL for the above XML data can be as follows:

```
<xs:complexType name="Wireline">
    <xs:annotation><xs:appinfo>
            <representation repType="binary"
                                    byteOrder="bigEndian"/>
    </xs:appinfo></xs:annotation>
    <xs:sequence minOccurs="number of points"
                    maxOccurs="number of points">
        <xs:element name="x" type="dfdl:binaryInt"/>
        <xs:element name="y" type="dfdl:binaryInt"/>
    </xs:sequence>
</xs:complexType>
```

The idea works best in a series of messages – a stream in Web Services. They have a data representation that has a WSDL schema and includes a SOAP message header that is largely unchanged (except for message label/number) throughout the stream in many cases. By separating representation and data, only the changing data and this in a high-performance binary representation is exchanged. The representation – DFDL structure of the data, and the unchanging parts of the SOAP message header are transported only at the beginning of stream. This is analogous to process in WS-SecureConversation where all messages in a stream share a security approach negotiated at the start of the stream.

3.2 Negotiation of Session Characteristics

The initial negotiation is a stage where two end-points exchange characteristics of the following stream where a stream for us is a special case of a session. For example, in a HHFR session between a mobile device and a gateway, the mobile device that is capable of Flexible Representation initiates *negotiation* with the mobile gateway. Then, Flexible Representation-capable HHMS Gateway receives the negotiation in a conventional SOAP message format over the HHMS transport. Subsequently, the gateway responds to the negotiation request with its capability in SOAP format. A capability response includes a data format for exchange, a reliable messaging scheme, and a security model. The agreed message format is used throughout the session. In addition to negotiating stream configuration, the header of stream is stored in Context Store. Currently, we use an ad-hoc scheme to store Context information. We intend to use the OASIS WS-Context [18] or any specification that the community adopts for such dynamic meta-data.

4. HANDHELD FLEXIBLE REPRESENTATION SOFTWARE ARCHITECTURE

In this section, we will briefly describe the software architecture of HHFR. Since binary messages are exchanged as a stream in the HHFR session, we need to change the representation of conventional Web Service specifications in areas like reliable messaging. Thus, we use customized HHFR schemes to ensure reliability and security model as well as SOAP Infoset encoding. At the beginning of a session, two end-points negotiate the desired quality of services (QoS) such as a reliable messaging scheme and security as well as an encoding/decoding format for binary messages. Fig. 1 is an overview of HHFR implemented as a part of HHMS in Web Services.

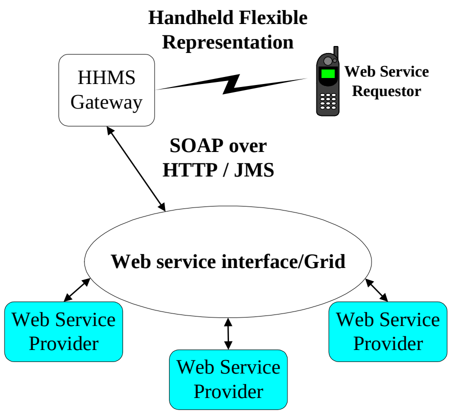


Figure. 1. Flexible Representation for mobile devices

4.1 Anatomy of Message in Flexible Representation

Message in our proposed binary form of SOAP Infoset achieves SOAP conformance using DFDL encoder/decoder and one-time transport of the SOAP message header. Later, the SOAP header, locally stored in gateway context server, is bound together with SOAP encoded body part to form a conventional SOAP message. The message encoding and decoding depend on a message format description file in DFDL. The message generation components are shown in Fig. 2. Each context block has a "pointer" to the place where context is stored (URI in the context store) that is different by each stream. The context has another "pointer" to the particular Flexible Representation used (URI - <xmlns:xs=http://grids.ucs.indiana.edu/FlexRep/>). It is similar to XML/SOAP namespace [19] and includes the description of data structure and negotiating scheme. Fig. 3 shows the context block.

HHFR uses two different level of message structure. The first is a low level application independent HHMS-level structure. The IPv4 [20] packet header is defined by bits, such as the first 4 bits for version. Our HHMS implementation defines a message format similar to the way IPv4 defines header. It defines the first integer as session ID, the second integer as state flag, and so on. Hence, the format can not be changed without new system deployment.

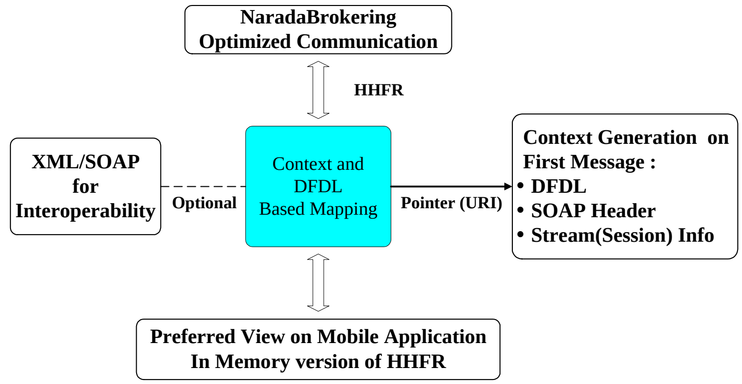


Figure. 2. Relationship of different forms of SOAP messages and their defining context

For the higher level structure, however, we use a descriptive way to represent the application SOAP Infoset as a binary message. A ContextHandler - a filter which utilizes DFDL reader/writer (encoder/decoder) - transforms a SOAP message or message capturing SOAP Infoset in any representation to a binary message format with the message format described in a DFDL file. During a message generation, a transformed binary SOAP message is put in a message block as a payload. Fig. 4 shows an example of message block in our design. Defining a message format by a description file provides a flexible way of changing message format in dynamic fashion that we can hardly expect from the base HHMS data encoding scheme. Of course processing a description file involves a possible parsing and validating error check.

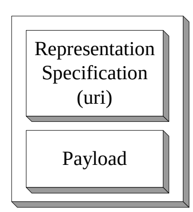


Figure. 3. Context block

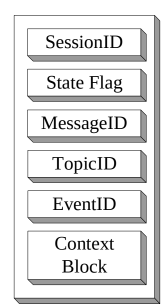


Figure. 4. Message block

Mobile clients of any application that could be bound to a JMS (Java Message Service) [21] or a NaradaBrokering [22] transport are capable of using HHFR. The current HHMS API provides a core subset of the JMS API. Therefore, applications where a message format has a payload and zero or more properties are naturally supported by our architecture. Subscribing to a JMS or NaradaBrokering topic and later unsubscribing defines joining and leaving sessions or starting and stopping streams.

4.2 HANDLERS

Handlers are system modules in the architecture that implement a context mapping (encoding/decoding message) and QoS issues. ContextHandler is implemented as a filter. It encodes and decodes messages based on the context in the context store. The conventional SOAP is one encoding option in ContextHandler. DFDL generates reader/writers from associated library for transforming representation between formats specified by the DFDL syntax file. A dynamic generation of mapping code is important to support general Web Services applications and their mobile applications without any new static code deployment.

WS-ReliableMessaging [23] is a quality of service specification from IBM and Microsoft. It is a transport independent SOAP specification and the implementation of this specification traces messages and assures delivery. We implement WS-RM on the message stream so that ReliabilityHandler traces each message block and ensures the guaranteed message delivery. The basic idea is optimized use of ACK (positive acknowledgement of receiving message) and NAK (negative acknowledgement of receiving message), reducing ACK and sending NAK when receiver detects a missing message ID. Each endpoint needs to have an RM processor and this must be compatible with overall reliable messaging so that the message sender can send message without prior knowledge about one type of receivers.

SOAP inherits few intrinsic weaknesses of XML [24], [25]. The new Web Service security proposal – WS-Security [26] describes how to authenticate two end-points to each other and to check the message integrity and message confidentiality. In our Flexible Representation, we have to be careful to apply XML level security - WS-Security - because there is no XML message format exchange throughout the session except the initial negotiation stage. We simplify a complex transaction model of WS-Security. The SecurityHandler use current existing Internet protocols and transport level encryption mechanisms to secure a session between the gateway and the mobile application. It participates in the negotiation and authenticates with HTTP-based authentication, such as HTTP Basic and HTTP Digest. And for the message encryption, the HTTP session use a HTTPS support from J2ME MIDP 2.0 [27]. For the non-HTTPS device, custom authentications and encryptions can be used with a lightweight encryption library [28].

5. EXAMPLE: A/V CONFERENCING IN MOBILE COMPUTING ENVIRONMENT

Global Multimedia Collaboration System (GlobalMMCS) is a prototype system designed to verify and refine our XGSP conference control framework. GlobalMMCS with mobile clients depicts a good example of our approach - the Flexible Representation for a SOAP Infoset and HHFR. Since its videoconferencing has utilized the idea when it is designed, the session integrated with HHFR is an example of a Flexible Representation system. In particular it uses a modification of the RTP data format (adding a topic pointer) as an optimized flexible representation.

Fig. 5 shows overall architecture of GlobalMMCS. GlobalMMCS prototype is built on the NaradaBrokering middleware. NaradaBrokering nodes route audio/video events to various communities and collaboration clients.

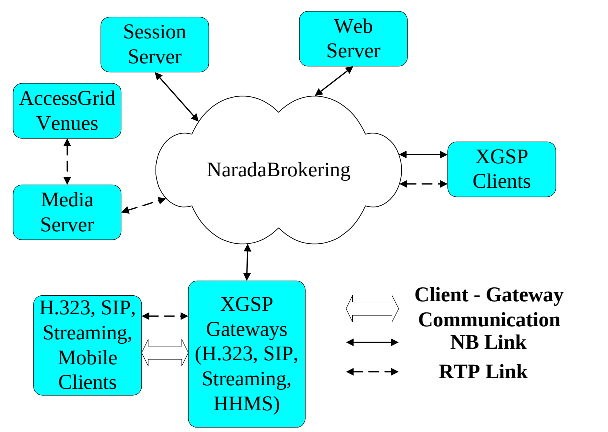


Figure. 5. GlobalMMCS architecture

A/V processing components such as video mixer, audio mixer and image grabber servers are developed using Java Media Framework [29] (JMF). We use the protocol stacks OpenH323 [30] and NIST-SIP [31] to implement the H.323 gateway and SIP gateway. The XGSP Session Server manages real-time A/V sessions. It receives messages from gateways and web server through different control topics on the NaradaBrokering. The XGSP web server provides an easy-to-use web interface to schedule meetings, join XGSP conferences and for administrators to perform the tasks of the system management. The NaradaBrokering infrastructure provides a scalable distributed messaging platform for RTP communications in these A/V collaboration applications. Any RTP client or server who wants to join XGSP session can subscribe/publish to the provided topics to receive and send A/V streams.

In order to integrate mobile devices to videoconferencing sessions the HHMS Gateway plays an important role as a Web Service interpolating between mobile and desktop Grids. Due to the limited capabilities of mobile devices, such as limited bandwidth, processing and memory capability, we cannot expect them to function with the power of a A/V client on a desktop PC and the HHMS Gateway provides the illusion to GlobalMMCS that the mobile clients are fully functional.

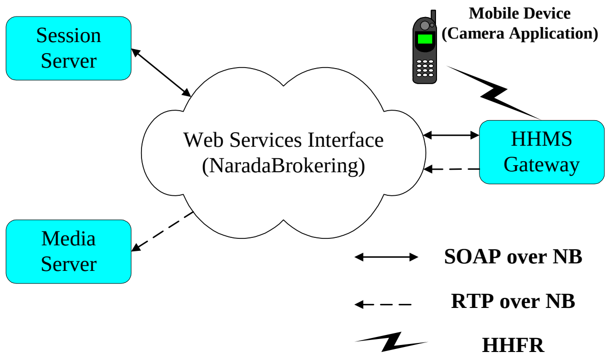
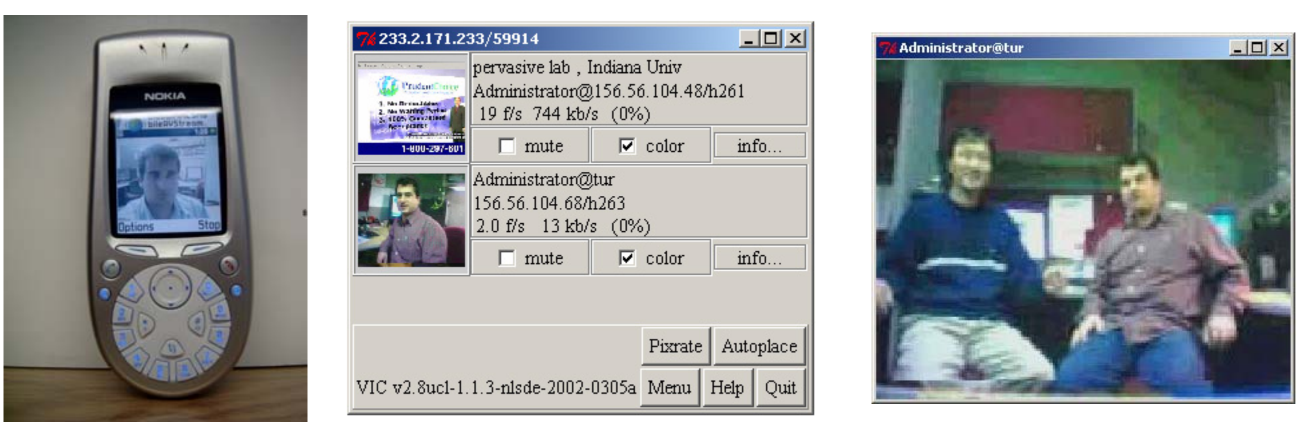


Figure. 6. Interactions between HHMS gateway and GlobalMMCS

A cell phone camera application for cellular phones can be provided as an example for this. Fig. 6 depicts a scenario that includes mobile device, HHMS Gateway and other components in GlobalMMCS. We developed a camera application to develop, implement and refine HHFR. We used Nokia 3650 [32] which has an embedded camera and MIDP 1.0 installed on it. Taking a picture and sending it to a server take several seconds. The image size is also 160x120 pixels. In order to send them to a videoconferencing session, we need to provide these pictures as continues video stream. We developed an application using JMF library to achieve image-to-stream conversion. This module also resizes the images by a factor of 2 before the conversion. The result video stream will be 320x240 pixels in size. The application then pushes the stream to the media server. The steps can be summarized as follows:

1. Camera application sends a SOAP message to HHMS gateway including ID of the session which client wants to join, including client ID. These are required to send JoinSession message to XGSP Session Server. HHMS gateway sends JoinSession message.
2. Upon receiving the reply from XGSP Session Server, HHMS Gateway replies to the camera application
3. Camera application takes pictures at predefined time intervals and put the image data inside message block as a payload. Session ID, client ID, topic ID and event ID are included as well. Event ID is used as a sequence number for images.
4. HHMS gateway takes the payload and forwards the image data to the corresponding image-to-stream converter module. The converter module is configured by HHMS gateway during the initialization step.
5. When the camera application wants to leave the session, it sets the state flag in message block to tell HHMS gateway that user wants to leave the session. In that case, camera application can leave the payload empty.
6. If the state flag is set, HHMS gateway sends LeaveSession message to XGSP Session Server.
7. Session Server releases resources dedicated on Session Server and Media Server after receiving LeaveSession



(a) (b) (c)

Figure. 7. A/V mobile prototype demo pictures

message from HHMS Gateway.

A similar scenario can be followed to send images from streams in the videoconferencing sessions to the mobile device. But this one requires a filter similar to XGSP Streaming Gateway to convert streams into a sequence of images.

Image shown in Fig. 7a shows a RealMedia stream played in RealPlayer on cellular phone. The original stream is a stream in XGSP session and the stream played on the cellular phone is converted by XGSP Streaming Gateway. Fig. 7b and 7c show the stream produced from the images captured by the camera application on the cellular phone. Fig. 7b shows it in VIC (Videoconferencing Tool used in AccessGrid) panel which shows all of the streams in that AccessGrid session. Fig. 7c shows only the stream produced from the images received from the camera application in a VIC frame. The captured images produced by the camera application on the cellular phone are converted into H.263 video stream format by an image-to-stream converter module developed using JMF library.

6. COMPARISON

Our Flexible Representation for SOAP Infoset tries to make more general and higher quality-of-service solution: W3C's MTOM [33] and XOP [34] preserve the basic structure of XML documents. It keeps tagged data model in the MIME format and only document contents are encoded in any binary format. Thus, it removes a data conversion problem of the conventional SOAP. It, however, imposes the same parsing overhead because it preserves hierarchical structure of its contents. We rather serialize SOAP/XML structure in binary format with tokens, but still keep data transparency with the data description file. Sun's Fast Web Services uses Abstract Syntax Notation 1 (ASN .1) [35]. Its schemas allow defining various binary or textual formats, still it doesn't use universal format of data model - XML schemas. Extreme! Lab shares the idea of negotiated binary protocol for scientific data with our Flexible Representation in an efficient way. Though, since their current implementation is focusing on high-performance scientific data processing, it is lack of general schema model for binary data format.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new paradigm of Web Services for mobile computing in collaboration systems. We describe our approach to SOAP Infoset encoding that preserves the SOAP semantic with a binary format for high performance message transmission and processing in collaboration systems while avoiding performance bottlenecks of SOAP, such as data conversion and parsing. The quality of services in the software architecture provides reliability and security model. Handheld Flexible Representation addresses specialized issues, like reliability and security model in intermittent and vulnerable wireless communications. Generally, HHFR scheme offers performance advantages in most of collaboration applications that is session based including Audio/Video Conferencing and shared Whiteboard.

Handheld Flexible Representation is currently partially implemented. The negotiation prototype is implemented over HHMS. Handlers are going to be implemented. The performance measurements are expected to show effectiveness of HHFR will be followed.

XGSP framework and its implementation GlobalMMCS enable multiple communities to collaborate with each other. Integrating mobile devices to GlobalMMCS expands collaboration systems to mobile environment. Mobile clients are able to send data (audio, video, etc.) to real-time collaboration sessions.

A dynamic resource discovery and a dynamic generation of Web Service client interface which would be generated by WSDL (Web Services description language) is out of scope of this paper. We assume that mobile clients have enough knowledge of Web Services to generate proper SOAP message. Currently, we use an adhoc scheme to store Context information. We intend to use the OASIS WS-Context or whatever the community adopts for such dynamic meta-data. Since a negotiation stage and a binary message reading/writing add certain overhead, we need a close investigation on the threshold of the architecture. This benchmark will lead us to the domain that we can get advantages of using Flexible Representation. Additionally, we need to investigate how much overhead introduced by adding QoS features with the same reason.

REFERENCES

[1] J. Kobielus, "Wrestling XML Down To Size: Reducing The Burden On Networks And Servers", *Business Communications Review*, Dec. 2003, p. 35-38.

[2] K. Chiu, M. Govindaraju, and R. Bramley, "Investigating the Limits of SOAP Performance for Scientific Computing", *Proc. of 11th IEEE International Symp on High Performance Distributed Computing HPDC-11 2002*, July 2002, p. 256.

[3] M. Govindaraju, A. Slominski, V. Choppella, R. Bramley, D. Gannon, "Requirements for and Evaluation of RMI Protocols for Scientific Computing", *Proc. of SC2000*, Nov. 2000.

[4] P. Sandoz, S. Pericas-Geertsen, K. Kawaguchi, M. Hadley, and E Pelegri-Llopart, "Fast Web Services", Aug. 2003, http://java.sun.com/developer/technicalArticles/WebServices/fastWS/

[5] World Wide Web Consortium, XML Information Set, Feb. 2004, http://www.w3.org/TR/xml-infoset/

[6] M. Beckerle, and M. Westhead, "GGF DFDL Primer", http://www.gridforum.org/Meetings/GGF11/Documents/DFDL_Primer_v2.pdf

[7] W3C Consortium, Report from the W3C Workshop on Binary Interchange of XML Information Item Sets, Sep. 2003, http://www.w3.org/2003/08/binary-interchange-workshop/

[8] J. H. Gailey, "Sending Files, Attachments, and SOAP Messages Via Direct Internet Message Encapsulation", Dec. 2002.

[9] D. Brutzman, and A. D. Hudson, "Cross-Format Schema Protocol (XFSP)", Sep. 2003.

[10] S. Oh, G. C. Fox, and S. Ko, "GMSME: An Architecture for Heterogeneous Collaboration with Mobile Devices", *Proc. of MWCN2003*, Oct. 2003.

[11] P. Eugster, P. Felber, R. Guerraoui, and A. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys, vol. 35, no. 2*, Jun. 2003, pp. 114-131.

[12] M. Caporuscio, A. Carzaniga, and A. Wolf, "Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications," *IEEE Trans. on Software Engineering, vol. 29, no. 12*, Dec. 2003, pp. 1059 – 1071.

[13] Oracle White Paper, "Oracle Web Conferencing Real-Time Collaboration System Architecture Overview", Oct. 2003.

[14] W. Wu, G. Fox, H. Bulut, A. Uyar, H. Altay "Design and Implementation of a Collaboration Web-services System", *Special issue on Grid comp. in Journal of Neural Parallel and Scientific Computations, Vol 12. No. 3* ,Sep. 2004

[15] G. Fox, W. Wu, A. Uyar, H. Bulut, and S. Pallickara, "Global Multimedia Collaboration System", *1st International Workshop on Middleware for Grid Computing*, Rio de Janeiro, Brazil, June 2003.

[16] AccessGrid, http://www.accessgrid.org

[17] Session Initiation Protocol (SIP), RFC 2543, http://www.ietf.org/rfc/rfc2543.txt

[18] M. Little at al, "Web Services Context (WS-Context) version 1.0", July 2003.

[19] World Wide Web Consortium, Namespaces in XML 1.1, Feb. 2004, http://www.w3.org/TR/xml-names11/

[20] Information Science Institute at University of Southern California, RFC 791, http://www.ietf.org/rfc/rfc791.txt

[21] M. Hapner et al, "Java Message Service Specification", version 1.1, Sun Microsystems Inc., Apr. 2002.

[22] S. Pallickara and G. Fox, "NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids," *Proc. of ACM/IFIP/USENIX International Middleware Conference Middleware*, 2003, pp 41-61.

[23] IBM, BEA, Microsoft, and TIBCO Software, "Web Services Reliable Messaging Protocol, Mar. 2004.

[24] B. Siddiqui, "Web Services Security", Mar. 2003, http://webservices.xml.com/pub/a/ws/2003/03/04/security.html

[25] B. Schneier, "Crypto-Gram News Letter - SOAP", June 2000, http://www.schneier.com/crypto-gram-0006.html

[26] IBM, Microsoft, and Verisign, Web Services Security, Apr. 2002, http://www-106.ibm.com/developerworks/webservices/library/ws-secure/

[27] Sun Microsystems, Mobile Information Device Profile (MIDP) http://java.sun.com/products/midp/

[28] Bouncy Castle, http://www.bouncycastle.org

[29] Sun Microsystems, Java Media Framework 2.1, 2001 http://java.sun.com/products/javamedia/jmf/2.1.1/index.html

[30] The OpenH323 Project, http://www.openh323.org/

[31] National Institute of Standards and Technology, http://dns.antd.nist.gov/proj/iptel/

[32] Nokia 3650, http://www.nokia.com/nokia/0,,2273,00.html

[33] World Wide Web Consortium, SOAP Message Transmission Optimization Mechanism (MTOM), Nov. 2004.

[34] World Wide Web Consortium, XML-binary Optimized Packaging (XOP), Aug. 2004.

[35] International Telecommunication Union, Abstract Syntax Notation One (ASN 1), http://asn1.elibel.tm