# A Demonstration of Collaborative Web Services and Peer-to-Peer Grids

Minjun Wang
*EECS Department, Syracuse University*
*Community Grid Laboratory, Indiana University*
*501 N Morton, Suite 222, Bloomington IN 47404*
minwang@indiana.edu

Geoffrey Fox
*Community Grid Laboratory, Computer Science Department, School of Informatics and Physics Department, Indiana University*
gcf@indiana.edu

Shrideep Pallickara
*Community Grid Laboratory, Indiana University*
*501 N Morton, Suite 224, Bloomington IN 47404*
spallick@indiana.edu

## Abstract

*Peer-to-Peer Grids is a new trend in scientific computing and collaboration. It is based on the Peer-to-Peer and Grids technologies and leverages the advantages of both.*

*We develop our collaborative PowerPoint and IE (Internet Explorer) applications with a common Peer-to-Peer Grid architecture. We make NaradaBrokering as our dynamic messaging environment and systematic use of Web Services as one type of our building blocks.*

*Making PowerPoint and Internet Explorer applications collaborative using their event Meta data and Instant Message as Web Services is useful in situations such as long distance education [2, 4] and web conferencing [6]. This is also a good demonstration of harnessing and leveraging of the power and richness of Web Services and Peer-to-Peer Grids computing.*

## 1. The Big Picture

We have developed collaborative PowerPoint applications, one of which is a master client that lectures and broadcasts its event messages to all participating clients. The participating clients of the collaborative PowerPoint receive and deal with the event messages, and render the process of the presentation individually. The applications including Microsoft PowerPoint and the resources of PowerPoint presentation files are deployed beforehand to each and every client so that control and communication of the whole process are message based. By using string-based event messages we improve the speed and efficiency of the collaboration performances because it lowers the Internet traffic greatly as compared to Shared Display, which is based on transferring image messages like bitmap.

Another type of participating clients is of IE (Internet Explorer) style, in which the IE is automated and the slides of the presentation are loaded and rendered inside the IE browser. This is suitable in situations where IE browser is preferable.

We use NaradaBrokering messaging service [14, 15] as the message environment to communicate event messages during the process. Together with the Network infrastructure, it correlates the elements of the Peer-to-Peer Grids, and therefore makes the collaboration possible.

Grids computing offers robust, structured, security services that scale well in pre-existing hierarchically arranged enterprises or organizations; it is largely asynchronous and allows seamless access to supercomputers and their datasets.

Peer-to-Peer is more convenient and efficient for the low-end clients to advertise and access the files on the communal computers; it is more intuitive, unstructured, and largely synchronous.

We have also developed and deployed Instant-Messaging Web Services to further improve the message communication and make it more portable and extensible in the Internet collaboration era.

We make the collaborative PowerPoint applications use the Instant Message as a Web Service by using XML (eXtensible Markup Language) and SOAP (Simple Object Access Protocol) protocol. We get and save the event message Meta data of the PowerPoint presentations, and make them as Web Services, so that the presentations can be rendered asynchronously as well as synchronously.

## 2. A Collaborative Web Service Model

We use a Shared Input Port Model [1, 3] for our collaborative PowerPoint and IE applications and web services, as in Figure 1.

In this model, the resource-facing input/output ports supply the information which is to define the state of the Web Service; the user-facing input, output ports pass control information by the user, and supply information for constructing the user interfaces. The messages on these ports define the state and results of each Web Service.

We have defined a protocol XGSP (XML General Session Protocol), which is an XML-based protocol to describe registration, session parameters, session membership, negotiation, etc. It defines session information for both general and the A/V subsystems [8, 9]. The collaborative PowerPoint and IE applications and web services use XGSP information to set up sessions with the session server.

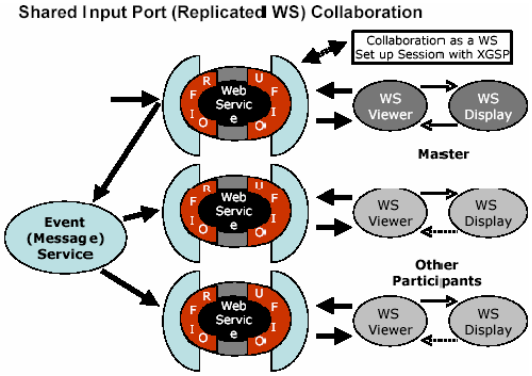


**Figure 1. Shared input port model for collaborative applications and web services**

Both master and other participating clients here have a copy of the PowerPoint application and presentations to be shared, and these PowerPoint presentation files are deployed in consistent directories between the master and the participants.

The master client captures the event messages, such as slide changes, window selection changes, etc., during its presentation of a currently open PowerPoint file. It sends out these event messages to all the participating clients, which in turn, navigate to the specific slide of a specific presentation, or to a specific shape/text range within a slide, based on these messages. This way, everyone can share the presentation or conferencing synchronously.

## 3. The Master Client

The master client plays a role as a Peer in the Peer-to-Peer Grids. It is the one that captures the event messages and sends them to the participating clients during its presentation of a session. It uses Automation, Connection Point object and Event sinks technologies [23, 24] in doing this.

Automation is a technology that enables the otherwise end-user applications to expose its functionality through interfaces, and the other applications can reuse the posted functions in its programs by using the methods resided in its wrapper classes.

In the master client of the collaborative PowerPoint, the client code controls the functionality of the PowerPoint application server through automation to make it started and visible at the beginning and to clear up at the end.

Microsoft has designed the Connectable Object technology that enable client and server object to communicate with each other in both directions. During the collaboration, when something interesting happened in the server object, it informs the client immediately in the form of a message, which is what we call an event.

The Connection Point objects are managed by the Connectable Object. This is where the outgoing interfaces are defined but their implementations are in the client event sinks. Each Connection Point is associated with only one outgoing interface. This is where the events occur and is therefore called the source interface for the client sink interface.

The sink is where the handlers of events are implemented, in other words, the event massages are handled and dealt with to generate different reactions.

This is illustrated in Figure 2.

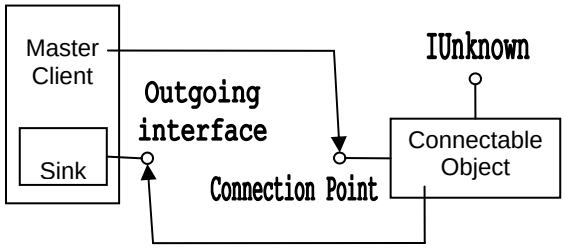


**Figure 2. Connectable object calls outgoing interface implemented by the sink. Master client handles events fired from the connectable object through the sink.**

During a presentation in the Master client, the PowerPoint event messages are sent through the Connection Point object to the sink object, where they are identified, processed by adding extra information about the master client, and sent out to the NaradaBrokering message broker, where the messages are distributed to the participating collaborative PowerPoint clients for rendering on the screen. In this way, the PowerPoint events are captured and dispatched.

## 4. The Participating Clients

When the Narada messaging broker receives event message from the Master client, it notifies the participating clients and broadcasts the message to them.

Each participating client plays a role as a Peer in the Peer-to-Peer Grids, and has its own copy of PowerPoint application and the presentation files about the topic it has subscribed to. The presentation files have been deployed to the corresponding directories as in host of the Master client. Each client processes the event messages independently.

When the client receives the message, it parses it and gets the different parts of information such as event type and its properties. It then dispatches the event type to the appropriate handler or method to process. The event type is the key to call different processing functions. The associated properties are used in the functions to generate the correct presentation results.

The client uses automation technology [22, 24] in rendering the session of a presentation. It calls the functions in a PowerPoint wrapper class under the instructions of the event message. The functions are actually mapped to the functions in the PowerPoint application. PowerPoint functions get called; do the tasks such as navigating through presentations and slides, and return the result values eventually to the caller functions in the wrapper class.

This is illustrated in Figure 3.

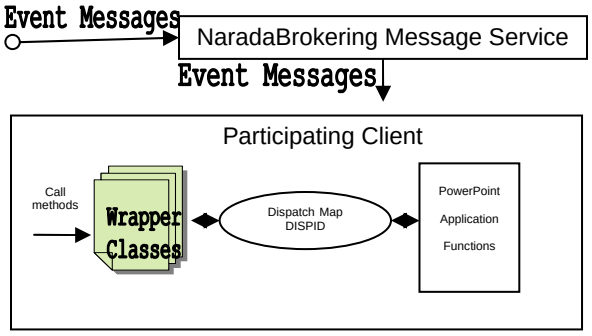


**Figure 3. The event messages invoke methods of wrapper class; the methods then map to functions of PowerPoint application through Dispatch Map/DISPID, execute and get result/status code back.**

The IE type of participating client functions in the same way as the one described here. It loads the presentation and renders each slide in its browser's screen.

Thus, the participating clients render the presentations being presented independently and simultaneously.

## 5. The Event Models

We abstract the event models of the collaborative PowerPoint and Internet Explorer applications to be of three levels of events: physical event, semantic event, and rendering event, from low to high, in that order.

The physical event is the event when a cursor is on a specific area of the screen, a mouse clicking, or a keyboard stroking, etc. When the master client is on a presentation session, the lecturer might use all combinations of the physical events to control the process.

The PowerPoint application converts these physical events to meaningful instructions to the applications, such as change slides, change windows, etc. These meaningful instructions we call them semantic events.

In our programs we make use of the Dispatch event interfaces of the PowerPoint application, connectable object, connection point technology and event sink to catch and deal with these semantic events. For some reasons, in Microsoft PowerPoint, one can only get the hexadecimal codes of these events instead of meaningful string name descriptions as in the other applications of the Microsoft office suites. With codes like this, one can not know the meanings of them and can not figure out which is which. We have done logical analysis according to the input / output of presentation processes, and, finally map each of the code to its corresponding meaningful string name in the event interface of the PowerPoint. We call this process a translation.

After getting them, the master client sends the semantic events through NaradaBrokering message service to the participating clients. The participating clients then call the functions of the PowerPoint through automation, according to each command of the semantic event, thus render the process of the presentation. We call this kind of event rendering event.

## 6. NaradaBrokering Message Service

NaradaBrokering is a messaging environment; it can be deployed as a Grid in Peer-to-Peer Grids, using robust, secure, structured and powerful machines and resources.

We integrate NaradaBrokering Message Service in our collaborative PowerPoint and IE applications to transmit event messages between clients. NaradaBrokering is a system that supports messaging in a Peer-to-Peer Grid [1,10]; it is a generalized publish-subscribe mechanism; it handles dynamic protocol choice, tunneling through firewalls; it supports TCP, UDP, multicast, SSL and RTP; it can run in client-server mode like JMS (Java Message Service) [15, 17] or in distributed Peer-to-Peer mode like JXTA [5,16]; it can be used in real-time synchronous collaboration like our collaborative PPT and IE; it has

replaced the JMS in the Anabas system of our implementation handling all collaboration modes.

NaradaBrokering system was written in Java language, and our collaborative PowerPoint and IE applications have been developed in C++. In order to communicate information between the two developing environment, we use JNI (Java Native Interface) as a tool to fulfill this task. The communication is a two-way conduit, both from C++ sending event messages to Java, and from Java to C++.

The Master client in our applications captures the event messages in PowerPoint and sends them to the NaradaBrokering message service system using the functions in JNI interface. In doing so, it first creates and embeds a Java Virtual Machine inside the C++ environment, maps data types between them, calls the JNI functions through the virtual machine.

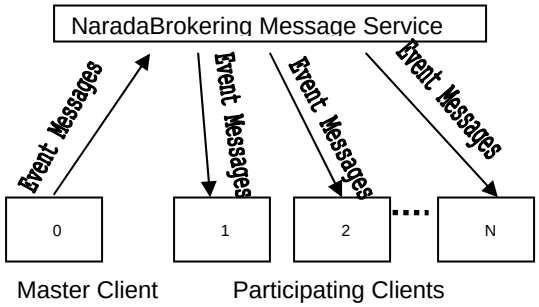


**Figure 4.**

As soon as the NaradaBrokering system receives a message, it broadcasts it to every participating client, using the notification mechanism, as shown in Figure 4. Here, the transformation of the message is from Java to C++ environment. The notifying method, i.e. onMessage(), is overridden to include native function calls to C++, so that the message type commands the appropriate C++ functions in the participating client application to perform the rendering process of the presentation. The functionality of the participating client is divided into C++ methods, and contained in a dynamic link library component (e.g. collabPPT.dll), which is loaded in the Java environment so that the Java native functions can make use of it. The JNI interface plays an important role in this communication direction.

Thus, the master and participating client of the collaborative PowerPoint and IE applications communicate and corporate with each other through the NaradaBrokering Message Service system.

## 7. Instant-Messaging and Event Meta Data as Web Services

Web Services along with Peer-to-Peer Grids play important roles in collaboration. Web Services enable developers to integrate functionality across businesses and organizations.

In our applications, we develop and make use of Web Services such as event Meta data and Instant Messages, so that the nature of the applications is of global collaboration.

The event messages from the applications can be marked up using XML tags, so that an XML document can be generated corresponding to the DOM (Document Object Model) format [20]. This DOM-based XML document can then be used as the unit of message communications between the clients of the Collaborative PowerPoint and IE services, it is transferred through the Internet using SOAP protocol, and it is the basis of Instant Message communication.

We have developed and deployed Instant-Messaging Web Services [13] for the communication in this project. The main services include function that markup event message as DOM-based XML document; function that get the event message out of the XML document, etc. The information of this web services such as its URI (Universal Resource Identifier) endpoint, its exposed methods, etc. are described in the WSDL (Web Service Description Language) file and then be deployed using this file. The users can find this web services using UDDI (Universal Discovery, Deployment and Integration), and then bind to the services they need and use them via the internet [21].

In the collaborative PowerPoint and IE applications, the master client send its event messages to the NaradaBrokering message services, which has discovered and bound to the Instant-messaging web services beforehand, which in turn make use of the exposed methods of the web services to make the plain messages received to be a DOM XML document, and then transfers and distributes the document via the internet to the participating clients for dealing and rendering.

The participating clients leverage the functionality of the Instant-messaging web services through the NaradaBrokering message service to get the plain event messages out of the XML document, and operate on the instructions of the messages syntactically, rendering the exact process the master client going through. This is shown in Figure 5.

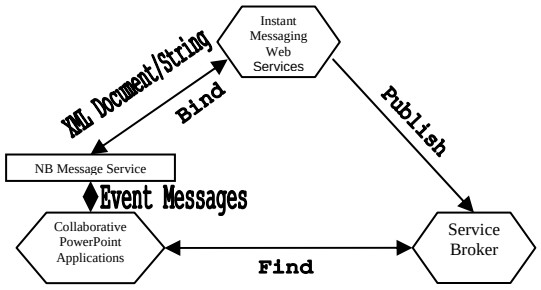


**Figure 5.**

We also capture and save the event message Meta data as files or in a database when the Master client is presenting and sending messages, and make it as a Web Service. This way, the presentations can be rendered asynchronously later by the subscribers, as well as synchronously as in the real-time session described in this text.

We develop another application that accesses the Meta data of the Web Service and renders the presentations; it does this in an on-demand way, according to the pace of the preference of the subscriber. The user goes through the sequence of a presentation on demand by clicking a button on the interface such as "Navigate". This way, the user reviews the stored presentations in his/her own way and need, taking the best advantages out of them.

This is meaningful in Distance Learning, Conferencing, and more.

## 8. Summaries

The Collaborative PowerPoint and IE applications integrate the master and participating client processes; cooperate with the NaradaBrokering message service; leverage the Instant-messaging and event Meta data web services. It can be used in distance learning, lecturing, conferencing, etc. With the advantages of its small text based message transferring, the robustness of NaradaBrokering message service, the efficiency of Instant-message communications, and the convenience of using event Meta data Web Services, it serves well in suitable areas, and it gives a good demonstration of the usage of collaborative Web Services and Peer-to-Peer Grids.

## References

[1] Geoffrey Fox, Hasan Bulut, Kangseok Kim, Sung-Hoon Ko, Sangmi Lee, Sangyoon Oh, Shrideep Pallickara, Xiaohong Qiu, Ahmet Uyar, Minjun Wang, and Wenjun Wu, "Collaborative Web Services and Peer-to-Peer Grids", presented at 2003 Collaborative Technologies Symposium (CTS'03)

[2] Collection of Resources on distance education by Geoffrey C. Fox. http://grids.ucs.indiana.edu/ptliupages/publications/disted/

[3] Fran Berman, Geoffrey Fox, and Tony Hey, "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons Ltd, Chichester, 2003. See http://www.grid2002.org

[4] Geoffrey Fox, "Education and the Enterprise with the Grid", Chapter in ref. 3

[5] Sun Microsystems JXTA Peer to Peer technology. http://www.jxta.org

[6] WebEx Collaboration Environment. http://www.webex.com

[7] Placeware Collaboration Environment. http://www.placeware.com

[8] Geoffrey Fox, Wenjun Wu, Ahmet Uyar, and Hasan Bulut, "A Web Services Framework for Collaboration and Audio/Video conferencing"; proceedings of 2002 *International Conference on Internet Computing IC'02*: Las Vegas, USA, June 24-27, 2002. http://grids.ucs.indiana.edu/ptliupages/publicatios/avwebserviceapril02.pdf

[9] Hasan Bulut, Geoffrey Fox, Shrideep Pallickara, Ahmet Uyar and Wenjun Wu, "Integration of NaradaBrokering and Audio/Video Conferencing as a Web Service". http://grids.ucs.indiana.edu/ptliupages/publications/AVOverNaradaBrokering.pdf

[10] Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko,Sangmi Lee, Shrideep Pallickara, Marlon Pierce, Xiaohong Qiu, Xi Rao, Ahmet Uyar, Minjun Wang, and Wenjun Wu, "Peer-to-Peer Grids", Chapter in ref. 3

[11] Sangmi Lee, Geoffrey Fox, Sunghoon Ko, Minjun Wang, and Xiaohong Qiu, "Ubiquitous Access for Collaborative Information System using SVG", Proceedings of SVGopen conference July 2002, Zurich, Switzerland. http://grids.ucs.indiana.edu/ptliupages/projects/carousel/pagers/draft.pdf

[12] Geoffrey Fox, Sangmi Lee, Sunghoon Ko, Kangseok Kim, and Sangyoon Oh, "CAROUSEL Web Service: Universal Accessible Web Service Architecture for Collaborative Application", November 2002, http://grids.ucs.indiana.edu/ptliupages/publications/Carousel_PerCom03.doc

[13] OASIS Web Services for Remote Portals (WSRP) and Web Services for Interactive Applications (WSIA) http://www.oasis-open.org/committees/

[14] Geoffrey Fox and Shrideep Pallickara, "The NaradaBrokering Event Brokering System: Overview and Extensions", *proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications* (PDPTA'02). http://grids.ucs.indiana.edu/ptliupages/publications/NaradaEventBrokering.doc

[15] Geoffrey Fox and Shrideep Pallickara, "JMS Compliance in the NaradaBrokering Event Brokering System", in the proceedings of the 2002 *International Conference on Internet Computing* (IC-02). http://grids.ucs.indiana.edu/ptliupages/publications/JMS-And-Narada.doc

[16] Geoffrey Fox, Shrideep Pallickara, and Xi Rao, "A Scalable Event Infrastructure for Peer to Peer Grids", proceedings of 2002 Java Grande/ISCOPE Conference, Seattle, November 2002, ACM Press, ISBN 1-58113-599-8, pages 66-75.

http://grids.ucs.indiana.edu/ptliupages/publications/ScaleableEventArchForP2P.doc
http://grids.ucs.indiana.edu/ptliupages/publications/foxwmc03keynote.pdf
[17] Sun Microsystems. Java Message Service. http://java.sun.com/products/jms
[18] Anabas Collaboration Environment, http://www.anabas.com
[19] Guy Eddon and Henry Eddon, Inside Distributed COM, Microsoft Press, 1998. ISBN 1-57231-849-X
[20] H. M. Deitel, P. J. Deitel, T. R. Nieto, T. M. Lin, and P. Sadhu, XML How to Program, Prentice Hall, 2001. ISBN 0-13-028417-3
[21] H. M. Deitel, P. J. Deitel, J. P. Gadzik, K. Lomeli, S. E. Santry, and S. Zhang, Java Web Services for Experienced Programmers, ISBN 0-13-046134-2
[22] Guy Eddon and Henry Eddon, Inside Distributed COM, Chapter 5, "Automation and Component Categories", Microsoft Press, 1998.
[23] Guy Eddon and Henry Eddon, Inside Distributed COM, Chapter 6, "Connection Points and Type Information", Microsoft Press, 1998.
[24] Microsoft Knowledge Base, http://support.microsoft.com/
[25] Web Conferencing Central's work on online meetings with PowerPoint, http://www.web-conferencing-central.com/web-conferencing-powerpoint.html