Components on the Internet

Panel Session

Jeffery Bonar, Nancy Lehrer, Kelly Looney, Stewart Nickolas, James Russell, Randy Schnier, Ted Selker

Abstract: The explosive emergence of the Internet forces us to rethink the traditional roles for client, server, and objects. Component technology has emerged as the way objects and frameworks are packaged for flexible object integration. In particular, components allow users who are not programmers to do their own integration. Up until now, components have mainly focused on object integration on end-users desktops and on a client: compound documents, application assembly, etc.

Existing component models like OpenDoc and Microsoft's OLE/OCX/COM provide function for compound documents, persistence, scripting, inking, and code management (registration). A series of Internet scenarios are emerging that demand new kinds of components. These scenarios are thin clients (Internet terminals), virtual reality/rich multimedia clients, collaboration, and business data access. Each panelist will explore one of these scenarios for it's requirements on an Internet component model.

Nancy Lehrer

The DARPA Sponsored Intelligent Integration of Information (I3) research initiative is concerned with accessing information from multiple heterogeneous sources and gathering this information into high-value end-user information products. The WWW is a natural medium as both an information resource and a means for packaging and organizing these hyper-linked information products. A distinguishing factor for high-value information products will not only be the types and sources of information with can be accessed, but be their ability to customize information products not only by consumer type, but also by consumer preference and consumer specifics.

Consider information packages modeled as complex object oriented queries such as "Get the catalogs for all diamond retailers in my state which offer settings valued more than $1000. It should contain their diamond setting catalogs for rings and pendants". In this query "catalog" is a complex object which includes certain attributes which also may be objects such as setting information, price structures etc. Additionally, each complex object is a mix of media types; text, images, video. Modeling a domain in terms of these objects is key to providing flexible information product capabilities.

We model information as object queries that can be tailored to typical sets of information consumers such as large jewelry retailer, small jeweler retailer or hobbyist based on the domain model. Each object query may have a default set of attributes which are interesting for that specific consumer group and a set of parameters which support customization based on user specifics. Local context query terms such as "located in my state" and user specific contexts such as preferred currency and distance units must supported. By combining advanced information integration techniques with the capability to model information queries based on a set of useful domain terms, information packages become significantly more useful.

Nancy Lehrer is a Senior Scientist at ISX Corporation, a small company specializing in associate systems. At ISX, Ms. Lehrer leads the Information Integration technology group developing tools and advanced technologies for accessing and integrating information from multiple heterogeneous sources and is very active in the DARPA sponsored research initiative Intelligent Integration of Information (I3). Ms. Lehrer has been involved in a range of activities relating object oriented representations to the issues of information access. She has conducted basic research in mapping object oriented domain representations to native data source representations and packaging information for specific types of users. Ms. Lehrer earned her Masters degree in Computer Science at the University of Massachusetts at Amherst in 1987.

Kelly Looney

After years of waiting, the Internet is providing the mainstream populace with it's second real dose of object technology. The first, of course, was the graphical user interface model of a desktop. We spent years with desktop trash cans as the object community's chief contribution to peoples understanding of computers. Now millions of computer users will be introduced to a new type of object, quite unlike the iconic folders and disks to which they have become accustomed. These objects represent code and data that has migrated across the Internet to land on their own computers. It has been estimated that over ten million computer users will have Netscape 2 on their desktops this year, every one containing a Java virtual machine. Microsoft has OLE on millions of desktops already. Truly, the Internet has finally brought object technology firmly into the mainstream, but once again mostly as a user interface technology.

The World Wide Web is currently, in many ways, a "Back to the Future" phenomena. Currently Web browsers have the distinctive feel of dressed-up 3270 block mode terminals. The corporate world will move to Web Applications in droves partly due to the fact that Web applications will feel familiar to this mainframe-oriented crowd. Many new applications built over the next few years will use the Web as a user interface. And the Web interfaces will become increasingly sophisticated over time due to things like Java and OLE. In part the applet or component is our new UI goodie, but it is also interesting in that it is portable. The object concept of modeling computer hardware to make software instantly portable has been floating around the object technology world for some time. Modeling is the real point.

But too much of today's discussion seems to be centered of new UI goodies. Objects need not be just the window dressing for this next generation of computer software. The Internet browser is our new way to interact with the world. But, what will we see? Objects can model the many domains that developers need to surface to users. The Internet can help bring objects back to their original simulation roots. We can show people new worlds to explore. There will be a tremendous demand over the next few years for ways to make legacy database information available. But these databases were typically designed by and for computer professionals. To make them comprehensible we need to build models that people understand. The real potential for objects to change peoples lives is in the way that objects can bring new worlds to our computer from anywhere and make these new worlds understandable.

In over three years with ParcPlace Mr. Looney has performed a variety of roles. He has been a consultant, an object methodology instructor and mentor, and most recently the Project Manager for the VisualWorks 2.5 release and the Release Manager for the VisualWave project. He is currently the product manager for the VisualWave product line, ParcPlace-Digitalk's Smalltalk-for-the-Internet development environment. Mr. Looney has a ten year background managing development efforts in the Simulation and Object arena.

Stewart Nickolas

Component scripting on the Internet and corporate Intranets will be a vital success factor for distributed applications. Object-oriented technologies have been employed by developers to decentralize large legacy systems into more manageable component based applications. This change in development methodology was necessitated by several factors. The life-cycle of an application has grown much shorter, measured in months not years. The fast pace of emerging and changing technologies creates difficult design and implementation considerations for development teams. Customers requirements have become much more demanding for applications. Customers have also become more willing to craft a final solution by integrating application components.

This move to components will be an evolution. Corporate customers will remain focused on forms based work-flow business applications using component technology to augment new generations of existing applications. Component technologies such as OpenDoc, OLE/OCX and Java applets will continue to facilitate the change from large applications to the more flexible component based applications.

Component based applications provide a level of integration and flexibility previously not available to the domain expert (i.e. someone who is not a programming expert, but an expert in their actual work domain). As component re-use becomes more of a reality, users will integrate and customize components to deliver dynamic applications. Scripting is an important part of customizing and extending components. Scripting enables users to combine components in ways which provide tailored solutions. The ability to customize components via scripting will be pervasive on both client and server. For example, the management of distributed object systems might be automated using scripts. Automated tasks include such things as deploying, monitoring, updating and load balancing distributed applications.

Key to the success of an effective scripting strategy is standards. Standards have been introduced as guidelines in component scripting and the user community will continue to evolve these initial guidelines. These standards are intended to maximize component interoperability and provide a consistent user experience. The role of scripting component based software will continue to expand providing a productive and efficient mechanism for programmer and user alike to craft the optimal solution for the task at hand.

Stewart Nickolas is a Senior Programmer at IBM. Stewart's main area focus over the past ten years has been component based visual builder environments. Stewart has spent the last year as an architect for the OpenDoc technology specifically focused on scripting. Prior to joining the OpenDoc team Stewart spent two years as the technical lead on a project creating a platform independent component based visual builder. In his eight years prior to joining IBM Stewart worked on visual programming environments and intelligent tutoring systems built using component based technologies which modeled knowledge acquisition for children learning rational number arithmetic.

James Russell

Traditionally, the dominant forms of distributed applications were developed with tools and technologies designed and marketed specifically for that purpose, such as OSF DCE, OMG CORBA, or (coming soon) Network OLE. However, recently the realm of distributed computing has changed dramatically with the rapid expansion of the Internet and World Wide Web. Now, literally tens of millions of people are engaging in distributed computing every day, simply by clicking their mouse in a Web browser. The ubiquity of this technology naturally leads to the realization that shortly all distributed computing will be done in this context. Very soon, we will see technologies integrated with the Web that will support the same function that we have in CORBA today, with CORBA and Network OLE existing only as special-purpose "legacy" components within this broader environment.

To some degree, this is possible today, but the tools are still crude. To facilitate this advance we need (and will see) application development tools and technologies that simplify the development of Web-based clients for interaction with both standard HTTP servers and legacy technologies. We will also see expanded, heterogeneous, Web servers that similarly support the function that today is associated with distributed object systems – combining HTTP access to objects as well as OMG (or other protocol) access. The result will be a World Wide Web that allows users to create, and inter-operate with, distributed applications with ease, in a much more open environment than we have seen before.

James Russell is a Research Staff Member at IBM's T. J. Watson Research Center, and is Senior Manager of the Object and Distributed Technology group there. His main area of research over the past several years has been in tools for developing distributed applications. Major projects include the Concert/C research prototype and IBM's DSOM product, for which he was one of the architects. Most recently, he has been focusing on the problem of developing applications exploiting Internet technologies. James holds a Ph.D. in Computer Science from Cornell University.

Randy Schnier

When client/server computing arrived on the scene, it held out the promise of reduced costs, better response time, and greater flexibility to respond to changing business needs. But today's two-tier client/server systems often come up short in delivering on these characteristics. Performance and response time degrade as more and more "fat" clients are connected to database servers, and the demands on client machine capabilities keep growing as more and larger applications are installed on each client.

Maintaining synchronized images of operating systems and business operations applications on all these clients is costly and complex, and it is difficult to construct architectures that work across a heterogeneous, multi-vendor environment, tending to lock customers into proprietary and inflexible single-vendor solutions.

The emergence of network-centric computing, incorporating the Internet and company "Intranets", has made it possible to more fully realize the benefits of client/server. By using a three-tier architecture with database servers in the top tier, business application servers in the middle tier, and lightweight clients in the bottom tier, customers can avoid the "fat client" syndrome and its associated problems. The business logic tier and client tier are ideal applications for the use of web objects, with objects that encapsulate business logic on the middle tier, and user interface/presentation objects dynamically downloaded to clients in the bottom tier. By combining platform-independent client technologies such as Java and web browsers with CORBA technology on the application server, it becomes much easier to deploy powerful, flexible, high-performance client/server systems.

Randy Schnier is an advisory software engineer at IBM's lab in Rochester, Minnesota. He has been a member and leader of a variety of software projects during his 12 years with the company, including design automation tools (EDA and CAD/CAM), IC fabrication process tracking, and commercial object-oriented middleware products. His current work involves developing lightweight OO client technologies for multi-tier client/server environments. Randy holds a Bachelor of Science degree in electrical engineering from the University of Wisconsin-Madison.

Ted Selker

I will present a number of rich user interface paradigms and describe the objects and components necessary to realize them over the Internet. First, a user's components need a richer model of how public they are. A wonderful cooperative work paradigm would allow a person like me to tell you everything I want to share – all the people I know, all the places I have been, all the things I can think of -- and I want them to be in front of you as we are having a conversation. What would that do – to have such a visualization available as people are making new computer and programming environments? What it would do is focus on access -- browsing and searching, and working towards having scenarios where the task is more showing and testing than searching and constructing.

Next, components much have a way to show peripheral context. The Room With a View (RWAV) is a room-mounted rather than head-mounted virtual-reality system, allowing an office worker to go back to using their peripheral vision to orient them as they do work on a separate focus screen. Although computer users rarely have large enough monitors to show an orienting amount of text on their screens, or bandwidth enough to download all the background information, they typically leave a backdrop of icons visible as an orienting context. Other orienting imagery of shelves with books and folders are projected on an office wall. Pointing at one of these 'books' or other icons on the wall opens it for manipulation. This paradigm allows one to use their peripheral vision to orient and foveal vision to concentrate, consistent with how the eyes are designed.

Finally, components need to have structures that allow them to be adaptive far beyond the simple caching now employed to speed Internet access. The COgnitive Adaptive Computer Help (COACH) is a system which records user experience to create personalized user help. It is an interface agent which teaches a user rather than acting in that user's behalf. For example, example information demonstrating how to perform a procedure is often valuable until the procedure is mastered. The COACH system records a user's experience and expertise for learnable things (syntactic and conceptual) as they are being used. COACH is an architecture for experimenting with adaptive user models. The content coached, the way COACH adapts, and the pedagogical paradigm can be changed by content definition tables, courseware facts, and learning rules.

Dr. Ted Selker is Manager of User System Ergonomics Research (USER) at IBM Almaden Research Center. Ted's group works on cognitive graphical and physical interface. Ted teaches at Stanford, and previously worked at Xerox PARC, *at Atari Sunnyvale Research Center. His recent successes include the design of the "TrackPoint II" in keyboard pointing device performance advantages derived from a special behavioral/motor match algorithm available in IBM computers and creating "COACH", an adaptive agent improves user performance.*

Jeffery Bonar

Jeffrey Bonar is the technical lead for IBM's Object Technology Products group in Austin, Texas. This group develops SOM, DSOM, CORBA Object Services and OpenDoc. He has been involved with a range of object technology through his career. He did basic research on application of objects to graphic and direct manipulation user interfaces. Based on that research, he founded a company that developed and shipped an object-oriented tool for corporate application development. Since joining IBM, he has focused on components, scripting, and general mainstreaming of object technology. He received his doctorate in computer and information science from the University of Massachusetts in 1985.