Architecture and Implementation of a Collaborative Computing and Education Portal

Geoffrey Fox
Florida State University
Computational Science and Information Technology
400 Dirac Science Library
Tallahassee FL 32306-4120
fox@csit.fsu.edu

1.Introduction

We have developed several web-based environments to support both education and computing during the last few years. Early examples were the web-based interface to the Landscape Management simulation system [28,29,30] and the use of Tango Interactive [10] for distance classes between Syracuse and Jackson State Universities. These were respectively essentially the first computing and educational web-based portals ever developed. ERDC provided they key initial support for this work, which has now blossomed in a nationwide activity. We have continued to work in both areas and here describe the essential features of a framework called ECCE for electronic communities for computing and education. ECCE builds on two technologies - the Gateway computing environment and the still developing Garnet collaboration system. ECCE embodies lessons from our past work and incorporates modern Internet technologies. It is built in a modular fashion to enable multiple applications to re-use the same basic technology. Further the design recognizes that is no longer possible for a University research group to build a complete system in this area; rather one must build a modular system that uses other academic, government or commercial components where possible. ECCE uses commercial technology for the event system, instant messenger, and the shared display collaborative module and allows a choice of commercial and research audio-video conferencing support. Only with such an approach can you build a sustainable high functionality system taking advantage of the latest technologies. These include distributed object and web technology with active self-defining objects. New standards such as XML, new languages like Java and new mobile hand-held clients are furthers enablers of web-based environments, which offer the promise of remarkable user productivity.

2. Requirements

We want to support the convenient interaction between multiple users (clients) and multiple resources. The resources are computers, networks, instruments, storage devices, and visualization systems for computing portals. They are curriculum pages, quizzes, homework, and registration systems for education. Both applications need information resources to provide the ever-increasing sea of knowledge in which either to learn to do research. We have learnt that information can be divided into two classes – structured and what we call the Gallimaufry below. For those familiar with Yahoo or Google, the structured information corresponds to a *directory* labeled by a thoughtful hierarchy of topics; the Gallimaufry is accessed the typical web search interface suitable for a hodgepodge of material. We build ECCE to support both classes of information using the W3C concept of a URI (Universal Resource Identifier) [35] to label all entities in the system. Note that a URI is just a general hierarchical label; it is does not have to be a web site. We can catalog all the computers in our organization using URI’s such as *organization://things/computers/pc/2001/serialnumber*. This objects would have properties attached it for which one would develop an XML schema – for computers, the schema would identify location, owner, manufacturer, CPU, memory, windows name, DNS name etc. Such metadata would typically (today) be specified in a XML file which is likely to have one or more web addresses – this is the URI for the metadata file – in general quite different from the URI of the object. In the education and training arena, the IMS and ADL activities [14,15] have specified the structured information with the expected hierarchical arrangement; they cover curriculum modules and student related information. Objects like grades and curricula pages and object properties like prerequisites and completion requirements are specified. We assume that a similar approach will be adopted in all areas with the key objects identified and schema for their properties developed. Furthermore we see the universal use of URI’s allowing an integrated approach based on a core XML schema specifying a hierarchical structure with specialized properties for each field. This is accessed with a generalization such as XQUERY (to XML) [34] of standard database methodology with all data (including the unstructured Gallimaufry with URI’s but no systematic arrangement) allowing a Web search like interface. Note the resources in the Gallimaufry and structured information directory are all labeled by URI’s. We expect the URI to be a useful classifier for the structured but not for the Gallimaufry.

Portals are built in terms of web access to services on objects and above we have discussed the underlying distributed object structure we will use. Note that we did not describe the particular model we will use – CORBA, COM, Java, SOAP – for this doesn’t seem to us critical. At the distributed object level one has services such as object lookup and registration, object persistence and database support, event and transaction services. Security and fault tolerance are key services needed by all portals. We expect the technology to be driven by commodity and commerce portals, which must also have an excellent Information service. Globus and Gateway have already developed good security capabilities optimized for computing portals and we can use these solutions. Object sharing and the collaboration service is an area we are concentrating on, as this very important in both application areas. For education and training, the collaborators are teachers and students while for computing they are researchers or perhaps computationalists and consultants. We have already found here that formal scenarios such as class or science lectures or project reviews can be successfully supported for synchronous collaboration where shared objects (documents) are displayed at different sites and their view kept consistent. More dynamic scenarios such as brainstorming or office hours need improved technology. Our collaboration research has focused on resource sharing and we have taken other key technology such as audio-video, chat rooms, white boards etc. from other projects. We have also learnt that quality archiving of real-time session and the integration of synchronous and asynchronous collaboration is very important. Most activities use person to person interactions as an enhancement to asynchronous interactions mediated by e-mail and other non-instantaneous information exchange. There are other services, which tend to be application specific. For computing one can list accounting, job monitoring, file services (such as FTP), visualization, programming, application integration or composition, seamless access for job submission, and a parameter specification service (e.g. get data from web form into your

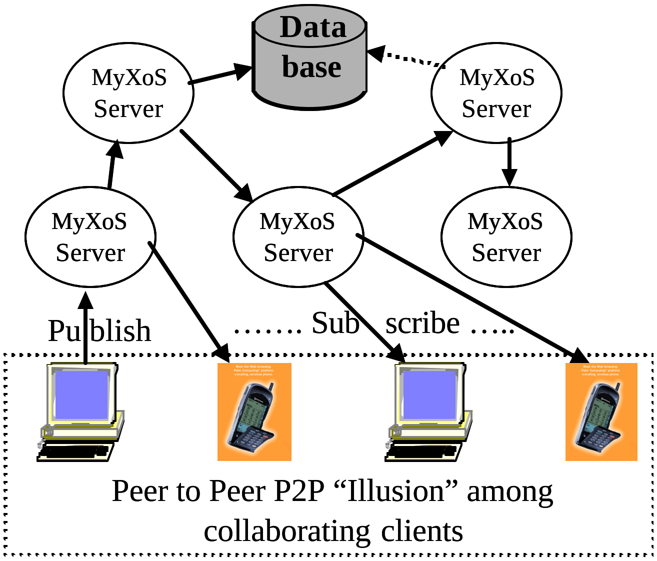
Fortran program wrapped as backend object). Also the basic application independent services need special optimization; in the computing case, we need high performance in object access and inter-object data transfer; we need to support special characteristics of our field such as information services that support mathematics and scientific data. Technically this corresponds to support of MathML, HDF5 [31] and higher-level specifications such as XSIL [24] and those developed within DICE (ICE) project at ARL [32]. For education, special services include delivery of lectures, support of "office hours", grading, homework submission, quizzes and assessment. These need to sensitive to IMS and ADL standards, which specify the structure of many of the relevant objects (tests and quizzes, assessment, grading sheets etc). ECCE has been designed so that the base framework can support generic services like security and collaboration and so that one can add these other services and other object standards.

Separately we have studied and prototyped in ERDC projects, the role of hand-held devices in both education and scientific research. Although these devices and their support is still immature, we see continued increase in the capability and use of these devices. Thus our portals must be designed so objects can be accessed from a wide variety of clients and further that such diverse clients can collaborate in the same session. This implies technically that we must separate object sharing from the rendering (display) of the object. This can be done with modern web systems where the object or information specification (application specific XML) is separated from its rendering (HTML, WML)

3. Technology Overview

Our approach has several key and novel features that have been designed to address issues coming from both previous research [1] and a detailed analysis [2] of major commercial tools in the collaboration and object management area. We will exploit the existing computing portal Gateway [3] that provides a web interface to access and manipulate computational science objects. The collaboration service is formulated as the Garnet system, which uses an integrated distributed object framework, consistent with the discussion in sec. 2, to specify all the needed object properties including both their rendering and their collaborative features. Sharing a complex object is difficult and we believe that one must have a sophisticated base object model to be able to build a successful broadly applicable collaboration system. Thus it makes sense to build the requirements of collaboration into any portal system. Including rendering information in an object's description allows one to customize to different clients and so build collaborative environments where one shares the same object between hand-held and desktop devices. We assume that we are building our computing portal on a computational grid infrastructure and so we can layer our high level services on top of the capabilities under development by projects such as the NEES Grid [4] being developed for earthquake engineering and for experimental physics, the Particle Physics Data Grid [5] and GryPhyN [6]. Users, computers, software applications, sessions, and all forms of information (including experimental data, simulation results and recordings of audio/video conferences) are all objects, which can be accessed from ECCE. Education objects are built to be compatible with IMS and ADL standards as explored in Sen's Syracuse PhD thesis [33]. This thesis was based on the course and grading support system ("The NPAC Grading System") used to support all our distance education including that with Jackson State. ECCE objects will all be self-defining; namely they will make explicit all the necessary metadata to enable ECCE to perform needed functions such as searching, accessing, unpacking, rendering, sharing, specifying of parameters, and streaming data in and out of them. This metadata is defined using a carefully designed XML schema GXOS and exploiting the new RDF framework from W3C [19]. Typically ECCE only manipulates the meta-objects formed from this metadata so that we build a high functionality middleware that only performs control functions. The XML meta-objects used by ECCE are proxies that point to the location of the "real object" they define and can initiate computations and data transfers on these real objects. Objects can be identified by a URI and referenced with this in either RDF resource links (such as <*rdf:description about="URI"* ..) or fields in the GXOS specification. The important URI's are the GXOS name such as *gndi://gxosroot/jpl/gem/users/?*, and the web location of either the meta-object or object itself. All objects in GXOS must have a unique name specified in a familiar (from file systems) hierarchical URI syntax.

ECCE software is largely written in Java (using Enterprise Javabeans in the middle tier) but Java/XML is only the execution object model of the meta-objects; one can load persistently stored meta-objects or control target "real objects" formed by flat files, CORBA, Microsoft .net (SOAP) or any distributed object system to which we can build a Java gateway. Our successful Gateway computational portal [1,3] has used this strategy already; here all object interfaces are defined in XML but CORBA access is generated dynamically. Further this system also only uses meta-objects in the middle tier and invokes programs and files using classic HPCC technology such as MPI. *This strategy ensures we combine the advantages of highly functional commodity technologies and high performance HPCC technologies.*



The Garnet collaboration service in ECCE uses the shared event model of collaboration where these events use the same base XML schema as the meta-objects describing the entities in the system. The uniform treatment of events and meta-objects enables us to use a simple universal persistency model gotten by a database client (shown in the figure below) subscribing as a client to all collaborative applications. Integration of synchronous and asynchronous collaboration is achieved by the use of the same publish/subscribe mechanism to support both modes. Hierarchical XML based topic objects matched to XML based subscribing profiles specified in RDF (Resource Description Framework from W3C) control this. Topics and profiles are also specified in GXOS and managed in the same way as meta-objects. These ideas imply new message and event services for the Grid, which must integrate events between applications and between clients and servers. This GMS (Grid Message service) is one major focus of our current computer science research. One extension of importance GMSME (GMS Micro Edition) handles messages and events on hand held and other small devices. This assumes an auxiliary (personal) server handling the interface between GMS and GMSME and offloading computationally intense chores from the handheld device. Currently we use JMS (Java Message Service) to provide publish/subscribe services for events in our prototype Garnet system but have already found serious limitations that we will address in GMS. [36] The event based synchronous collaboration model handles both the classic microscopic state changes (such as change in specification of viewpoint to a visualization) but also the transmitted frame-buffer updates for shared display which our experience has shown to be the most generally useful sharing mode for objects. We also support shared export where objects are converted to a common intermediate form for which a powerful general shared viewer is built; shared PDF, SVG, Java3D, HTML and image formats are important export formats. Collaborative visualization using either shared SVG or Web export is a promising capability supported by ECCE.

Although the use of XML based objects is relatively well understood, there appears to be less consensus as to the distributed programming or execution model needed to build the services needed by applications. In other words, what is the distributed operating system for the objects and meta-objects? The Ninja project [7] at UC Berkeley is addressing such issues with a philosophy similar to our approach, which is termed MyXoS, and supports ECCE with such capabilities as the creation, access, copying and editing of meta-objects. MyXoS has a "shell" similar to that provided by UNIX but specified (at a low level) by RDF statements and aimed at manipulating GXOS objects not files in UNIX. W3C likes to talk about the Semantic Web [8] formed by the synergistic interaction of web resources and this intriguing concept is an underlying research issue for systems like Ninja and MyXoS. Another important trend is peer-to-peer computing (P2P [24]) with recent work typified by JXTA from Bill Joy at Sun Microsystems [9]. As shown in figure above, collaborative systems create P2P networks although in our approach (and most other systems), this is an "illusion" for the P2P environment is created by the routing of messages through a network of servers. Here another interesting research issue is how best to perform this mix of software and hardware multicast and where the servers should be placed; MyXoS allows the dynamic instantiation of servers to support clusters of clients with similar subscription profiles. The message routing strategy needs to integrate the published topic and subscription profile objects and is quite complicated for heterogeneous client subscriptions.

The ECCE prototype will only implement basic versions of these ideas and the extensions needed for particular scientific fields or particular educational and training applications imply major research and implementation challenges. Research areas include the Grid Message Service as well as the Grid server architecture implementing the "illusion of P2P". We will also research the structure of MyXoS and the management of billions of meta-objects in an efficient effective fashion. Indeed, the question "is this a reasonable model at all with separate object and meta-objects" - will be an important lesson from our research. Our use of RDF as the scripting language of the Semantic Web will challenge this relatively simple meta-data model and perhaps point the way to improvements. Integrating the event models between the different services and synchronous and asynchronous collaboration modes is another hard area. We will build in general support for shared export but each case is non trivial and implementation will need substantial interaction with users and careful software design. There is a major effort to understand computational portals within the Grid Computational Environment working group of which Fox is a co-lead. This Grid Forum [23] activity is working with 16 different portals and describing them with a common template [17]. We will continue to work with this international group so as to ensure that we feed important lessons into the Grid architecture and that we bring to our systems the best ideas for distributed computational environments.

For any application domain, a major task will be to establish the formal specifications and procedures necessary to integrate all archived and derived information (distributed objects) into a common access framework. This implies extensions to the current GXOS XML Schema to handle the specific resources for particular application domains. Our overarching goal is to incorporate collaborative object technology into a Grid-based, problem solving framework that gives groups of researchers the ability to conduct and manage in a scalable fashion complex, distributed, large-scale simulation and data analysis problems. Thus, in the computing arena, ECCE seeks to dramatically improve the end-user environment for managing the needed collaboration, data access, and computational tools that researchers need to conduct scientific research. In education, ECCE provides a framework for a new class of learning environments integrating capabilities now seen separately in systems like Blackboard, WebCT, Centra and WebEx.

In the following Appendix, we give details of the core underlying technologies for Garnet, ECCE and MyXoS.

i.Appendix

A.1 Synchronous Collaboration

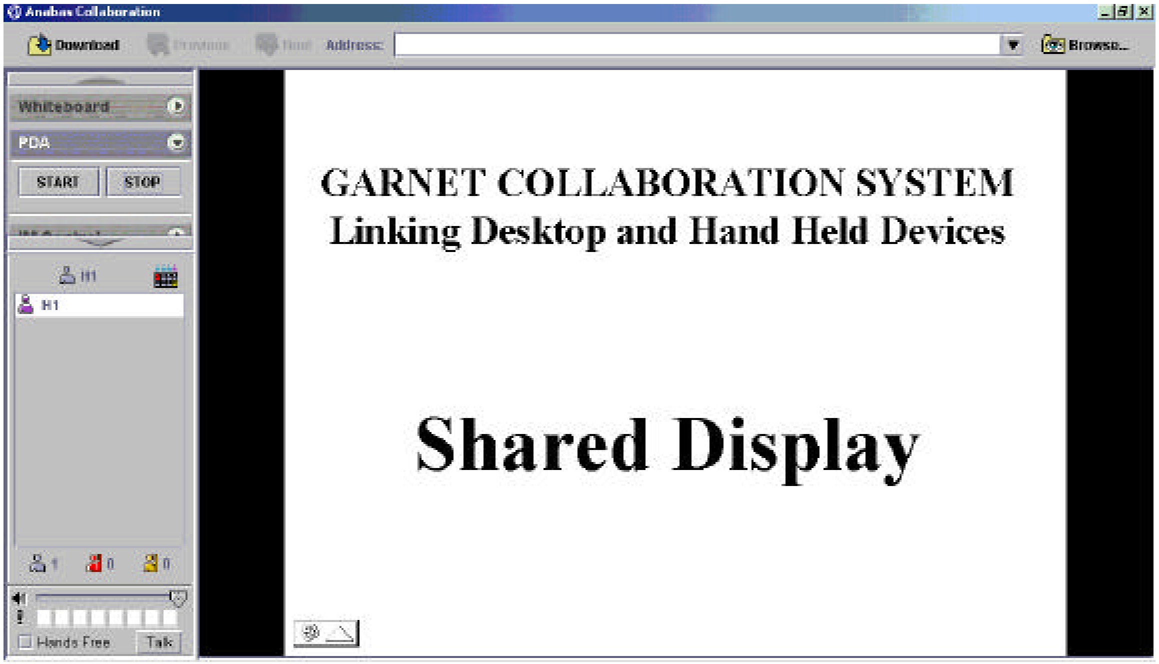
In this and following subsections, we define the key concepts and components of our proposed system. This is done in a glossary fashion for the broad categories of synchronous collaboration, portals and environments and distributed object technology.

A.1.1 Synchronous Collaboration Capabilities

This refers to object sharing in real-time with events recording state changes transmitted from a "master" instantiation to replicas on other clients in same session. Fox at Syracuse produced a research system of this type *Tango* [10] and includes lessons from this in Garnet. As detailed in the FSU survey [2], the three leading commercial systems, *Centra Placeware* and *WebEx*, are quite similar to themselves and to Tango. Such systems typically support:

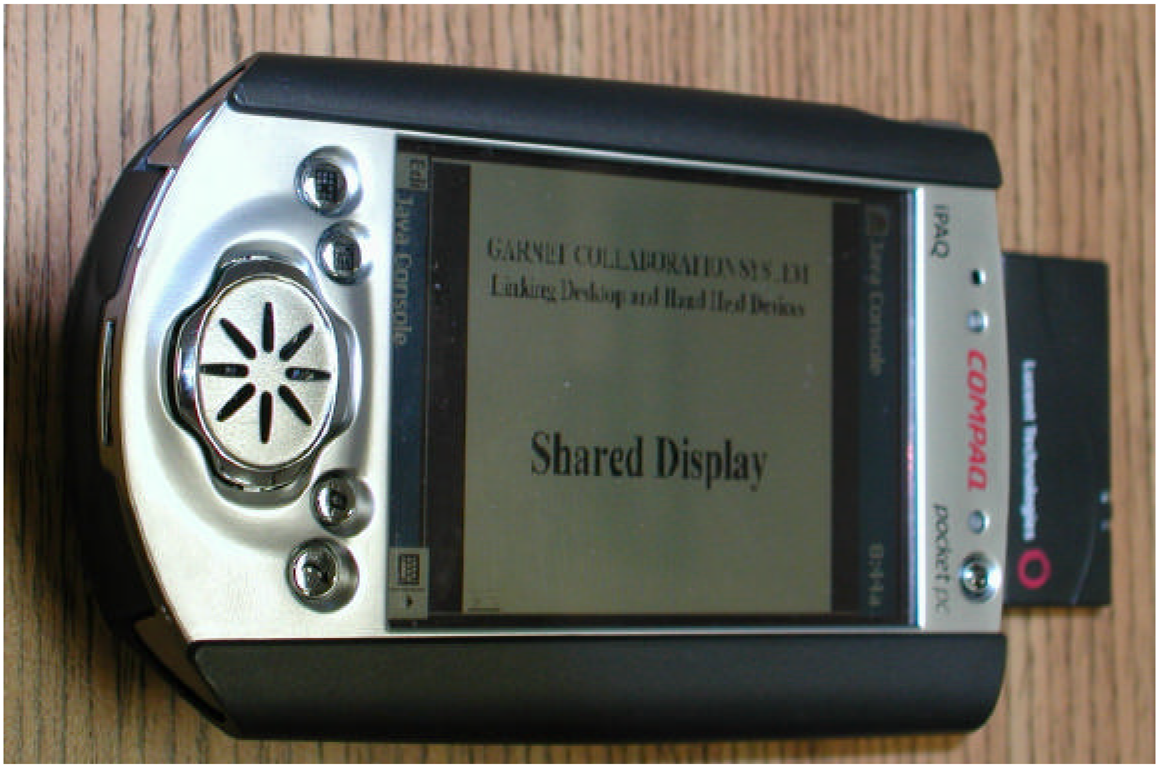
- Shared documents using either shared event, shared export or shared display. Note "document" here includes visualization, web page, Microsoft Word etc.
- Text Chat/Instant Messenger/Polling/Surveys/Attention getting tools
- White board and annotations (transparent white board) of shared documents
- Audio-Video conferencing
- User registration
- Recording Session

The Garnet system has these capabilities using either HearMe or Access Grid for the conferencing function. This type of capability has applicability to areas like real-time earthquake analysis, crisis management and also to the virtual control room for space missions. It has been most successfully applied to business briefings or distance education. Fox successfully employed Tango in this fashion for a set of courses given in Jackson State University in Mississippi from 1997 onwards with him as teacher at Syracuse or Florida State.



GARNET COLLABORATION SYSTEM
Linking Desktop and Hand Held Devices

Shared Display



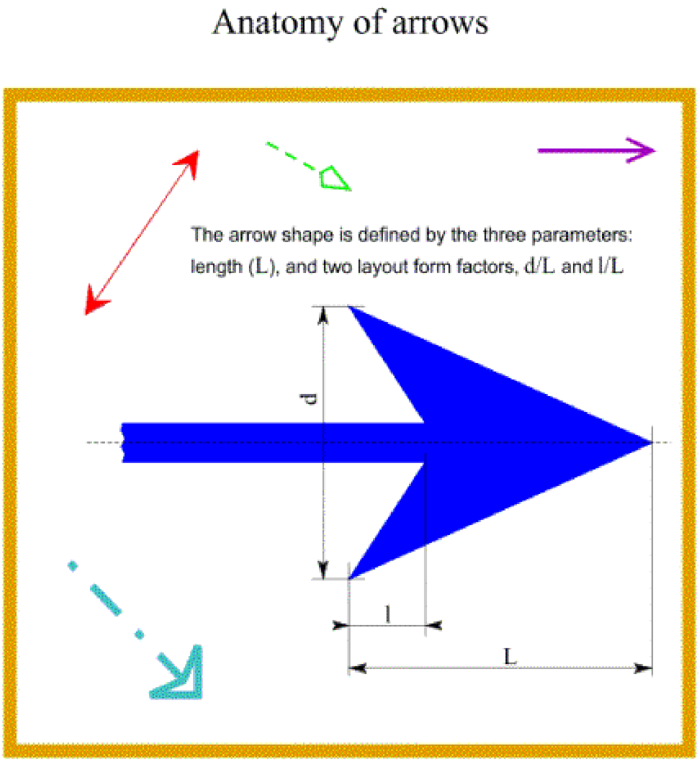
A.1.2 Shared Display

Shared display is the simplest method for sharing documents with the frame buffer corresponding to either a window or entire desktop replicated among the clients. Modest client dependence is possible with PDA's for example receiving a reduced size image. Some collaboration systems support remote manipulation with user interactions on one machine holding a replica frame buffer transmitted to the instance holding the original object. This is an important capability in help desk or consulting applications, similar to situations that occur frequently in the debugging of code. As this works for all applications without modifying them, this is the basic shared document mechanism in Garnet. The public domain VNC [11] and Microsoft NetMeeting were two of the earliest popular collaboration systems to implement this capability.

A.1.3 Shared Export

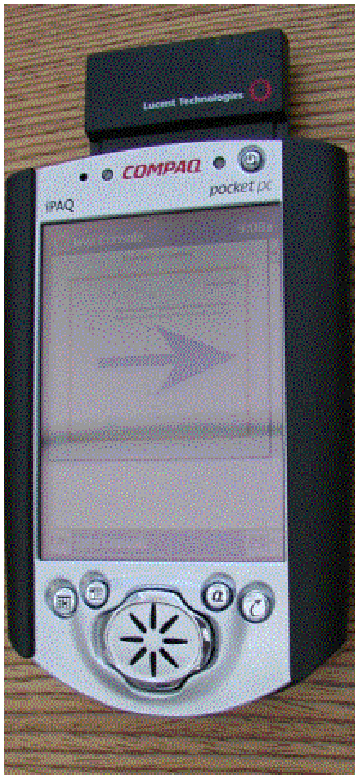
Shared display does not allow significant flexibility; for instance different clients cannot examine separate viewpoints of a scientific visualization. More flexible sharing is possible by sharing object state updates among clients with each client able to choose how to realize the update on its version of the object. This is very time consuming to develop if one must do this separately for each shared application. The shared export model filters the output of each application to one of a set of common formats and builds a custom shared event viewer for these formats. This allows a single collaborative viewer to be leveraged among several different applications. WebEx uses a shared virtual printer which is achieved with shared Acrobat PDF export in Garnet. The scalable formats SVG and PDF are particularly interesting and support of collaborative viewers for them is a major advantage of Garnet. Scalability implies that each client can resize and scroll while preserving correct positions of pointers and annotations. SVG is useful as it is already available for Adobe Illustrator and we can expect both PowerPoint and Macromedia Flash to be exportable to this syntax. Currently there is a Flash (which is a binary 2D vector graphics format) to SVG converter from the University of Nottingham; Office 2000 (save as web page) already exports PowerPoint to VML - an early proposal for the W3C SVG process.

We would recommend building SVG exports into tools like whiteboards and 2D scientific visualizations to allow convenient interchange among these different 2D presentation tools. We can expect Java3D and X3D [12] to allow similar general collaborative viewers to support collaborative 3D visualization. One should perhaps look at these 2D and 3D standards for the geographical information systems (GIS) area this would enable collaborative map-based displays using commodity technology like shared SVG viewers.



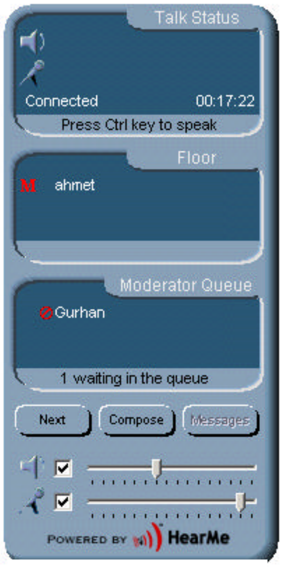
Anatomy of arrows

The arrow shape is defined by the three parameters: length (L), and two layout form factors, d/L and l/L



A.1.4 Access Grid

The Access Grid [13] is a very successful community audio-video conferencing system developed by Argonne National Laboratory. We support this in Garnet although we substitute (augment) its shared PowerPoint capability with shared document capability from Garnet and the commercial synchronous collaboration systems. The Access Grid could be a useful capability for "central" sites with lower end systems like HearMe used broadly.



A.1.5 HearMe

HearMe [14] is a leading commercial Internet desktop audio conferencing service supporting both PC and telephone client with archiving and replay. We have installed a HearMe system at FSU. Note that audio quality is a critical problem for Internet collaboration, as audio needs negligible bandwidth but excellent quality of service, which is often not available. Thus we use a system that allows phones as an integrated backup - note HearMe archives all audio whether it is from phones or purely Internet based. The archived audio can be replayed using streaming formats (such as RealAudio) with the W3C SMIL syntax integrating this with shared documents. We are investigating desktop video solutions but experience has found this not as critical as audio and so it is currently lower priority;

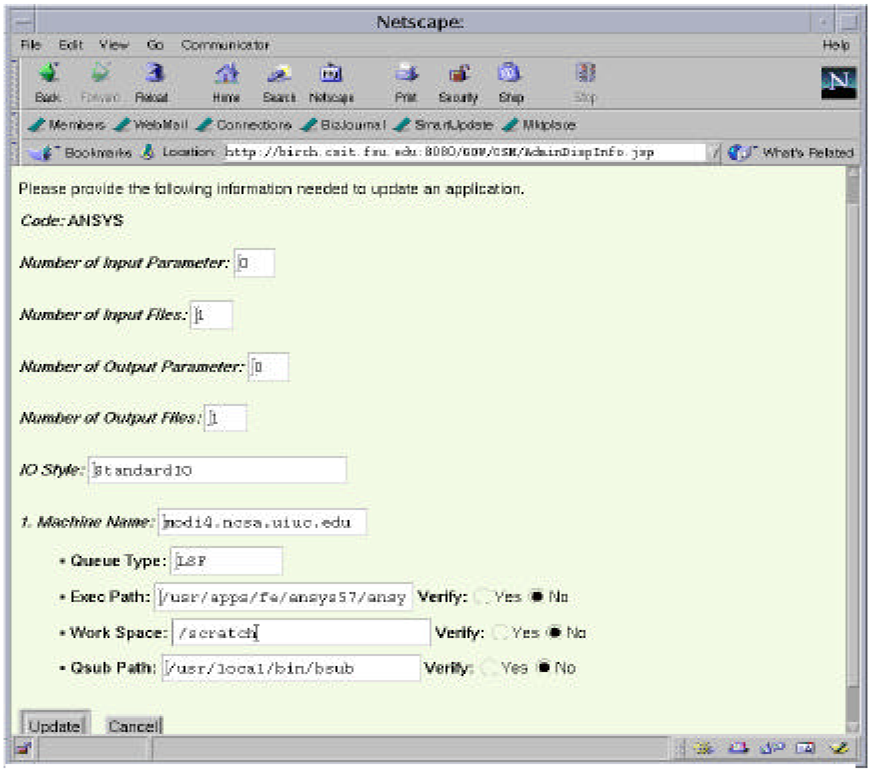
we will not develop this ourselves but use the best academic or commercial practice. We intend to study over the summer, ways of using the SIP and H.323 (two standards for conferencing tools) compatibility of HearMe to bridge it to Access Grid. This should allow desktop users to link to Access Grid sessions in a convenient fashion.

A.2 Portals and Collaborative Environments

A.2.1 Education and Commodity Portals

WebCT [26] and *Blackboard* [27] are leading education portals and are typical of managed information portals. The GXOS schema extends ideas present in the IMS [15] and ADL [16] initiatives for "learning object standards". We can expect conformance to these standards to allow exchange of course material between different management systems. Actually GXOS does not agree in detailed syntax with these standards but rather has a Schema, which allows GXOS objects to be mapped (by XSLT) into the IMS and ADL standards. For instance GXOS views education specific structure as an extension to a framework designed for general meta-objects, messages and events. IMS and ADL take an education centric view.

We have also examined the structure of commodity portals such as Yahoo and Excite and the structure of the news sites from CNN, New York Times etc. These are supported by the hierarchical topics (channels) in GXOS with customization using user profiles in GXOS; we call this part of GXOS *portalML*. We see these broad portal activities as important as providing guidelines so that more specialized scientific research environments can be constructed in ways that best leverage "COTS technologies". A FSU-JPL collaboration has proposed developing a suite of XML standards and has developed a preliminary DTD to cover the earthquake area for both simulation or observation based sensor and field data


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Members WebMail Connections BizJournal SmartUpdate Mktplace
Bookmarks Location: http://birch.csit.fsu.edu:8080/GSW/CSM/AdminDispInfo.jsp What's Related

Please provide the following information needed to update an application.

Code: ANSYS

Number of Input Parameter: 0

Number of Input Files: 1

Number of Output Parameter: 0

Number of Output Files: 1

IO Style: StandardIO

1. Machine Name: modi4.ncsa.uiuc.edu

- Queue Type: LSF
- Exec Path: /usr/apps/fe/ansys57/ansy Verify: ○ Yes ● No
- Work Space: /scratch Verify: ○ Yes ● No
- Qsub Path: /usr/local/bin/bsub Verify: ○ Yes ● No

A.2.2 Computing Portals

Computing portals provide web-based computing or problem solving environments. 16 recent projects of this type have been gathered together by the Grid Forum Computing

Environment working group [17]. This includes the FSU Gateway activity [3], which is to be integrated into Garnet, initially using shared display and shared Java server pages. Computing portals provide the means for managing the simulation and data analysis tasks for a group of distributed computational collaborators.

A.2.3 Collaborative Portal

A collaborative portal is a system that provides a portal (a web-based access to a particular application and/or set of Internet resources viewed as distributed objects) combined with ability to share accessed objects. This sharing includes synchronous and asynchronous access with the latter involving "channels" or "bulletin boards" or gotten by just posting a web page and informing interested parties in an informal fashion. Both portals and collaboration requires high-level metadata about the accessed objects and involved users. Thus we combine both concepts in ECCE with a single supporting information management service.

A.2.4 Garnet Collaborative Service

Garnet is the research collaboration system embodying ideas described in sections 2 and 3. Key functions include the integration of synchronous and asynchronous collaboration both in terms of topic publishing (channels) and object management. Thus it combines capabilities of synchronous collaboration and portals like *Gateway*, *WebCT* or *Blackboard*. It uses the *Access Grid* or *HearMe* systems for conferencing. It supports hand held devices, archiving and replay of collaborative sessions. Image (JPEG, GIF, PNG) SVG PDF and Java3D shared export viewers are planned. The latter three formats are scalable and support separate viewpoints (zooming) on each client. Garnet is designed in a modular fashion with clean interface for collaborative applications, which use the common GMS mechanism and GXOS Schema for exchanging state update events.

A.2.5 ECCE Electronic Community for Computing and Education (or Education and Computing Collaborative Environment)

ECCE is the digital networked environment described here that supports education, science research and computation and is based on *Garnet*, *Gateway* and *MyXoS*. ECCE is a collaborative portal using the Garnet collaboration service.

A.3 Distributed Object Technology

A.3.1 ECCE Basic Objects

These are smallest unit with which information is recorded and considered as a separate object. The result of one sensor at one time is such a basic object. Formally these are leaf nodes of structured data in a GXOS tree.

A.3.2 ECCE Gallimaufry

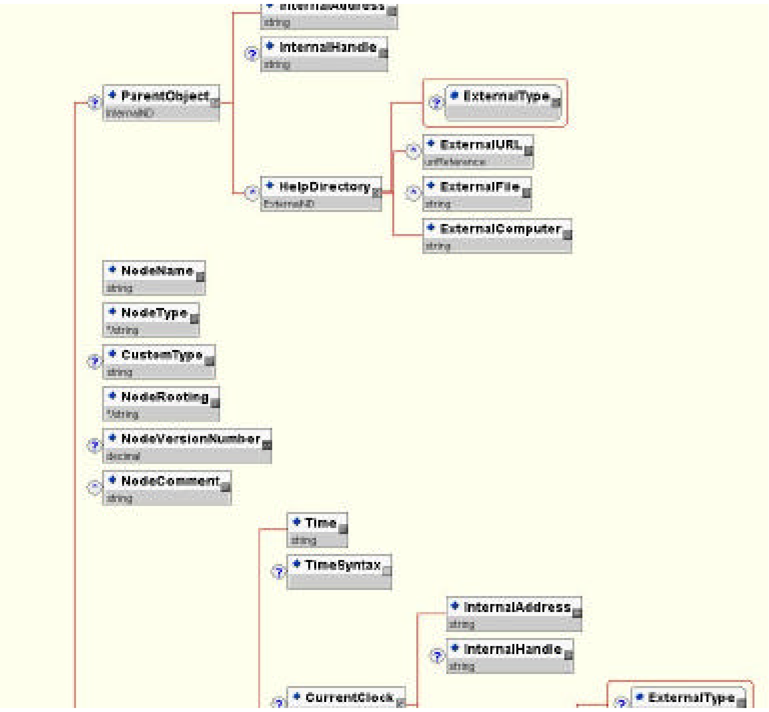
The Gallimaufry (hodgepodge or jumble) is the heterogeneous collection of ECCE knowledge made up from a multitude of sources including electronic mail, reports, presentations, data analyses, archived visualizations and meetings. It explicitly excludes the very structured and numerous sensor data or any information aggregate that is best looked at in a structured fashion.

A.3.3 ECCE Aggregates

Aggregates are collections of either basic objects or Meta-objects, which are usefully considered as a single unit - often because a group of basic objects are stored together in a single file. In GNDI, an aggregate is the collection of all Meta-objects stored as children of a GXOS node. An aggregate is defined in GXOS as a general sub-tree. Examples of aggregates might include the data produced by a given sensor over time where each reading is a basic object. We use aggregates to join related objects together into larger objects and so reduce the total number of meta-objects that users and MyXoS must explicitly manipulate.

A.3.4 GXOS Garnet eXtensible Object Specification

GXOS is an object specification realized as a collection of XML Schema in a single namespace defining a general hierarchical data structure where each node supports extensions to define different application domains; Users, Security, Computers, GMS, IMS/ADL (Education) and Computing (Gateway in a particular application domain) are particular extensions. We can view GXOS as having three basic capabilities expressed with same overall structure and three different sets of extensions; there is *resourceML* (defining the base objects like users, documents, computers); *portalML* describes the virtual environment with topics, user profiles, client renderings; *GMS* describes the messages that communicate between the subsystems.



A.3.5 Meta-object

Meta-objects are the basic units of GXOS, and they can be either at leaf or internal nodes of a GXOS tree. Meta-objects typically only contain Meta-data and use the GXOS Object Realization Schema type to specify access to the "real object". There are three ways an object can be related to a meta-object. Small objects such as basic objects or ECCE events would be self-contained i.e. the GXOS schema specifies the object and there is no distinction between meta-object and object. Secondly the object can be specified outside GXOS but its realization can be internal to GXOS (e.g. an RDF literal data type). Finally GXOS can reference a specification in any distributed object framework such as Java, CORBA, .NET or general SOAP protocol. One of the most common ECCE objects, large visualizations or sensor data collections would be likely to be stored in a hierarchical tape storage system and fall into the third category. Note that we will specify all properties of a sensor measurement in GXOS but manipulate in aggregate form with the metadata just summarizing the information. Some parts of the analysis will want to generate the native XML version of an basic object and MyXoS supports the multi-resolution view of information - one just needs to specify which of the tree you wish to look at. Note all nodes of a GXOS tree whether internal or leaf, have metadata and can be viewed as meta-objects.

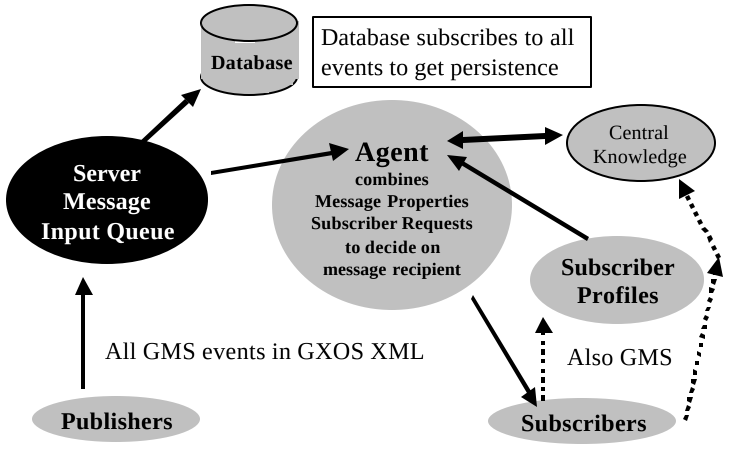
A.3.6 GNDI Garnet Naming and Directory Interface

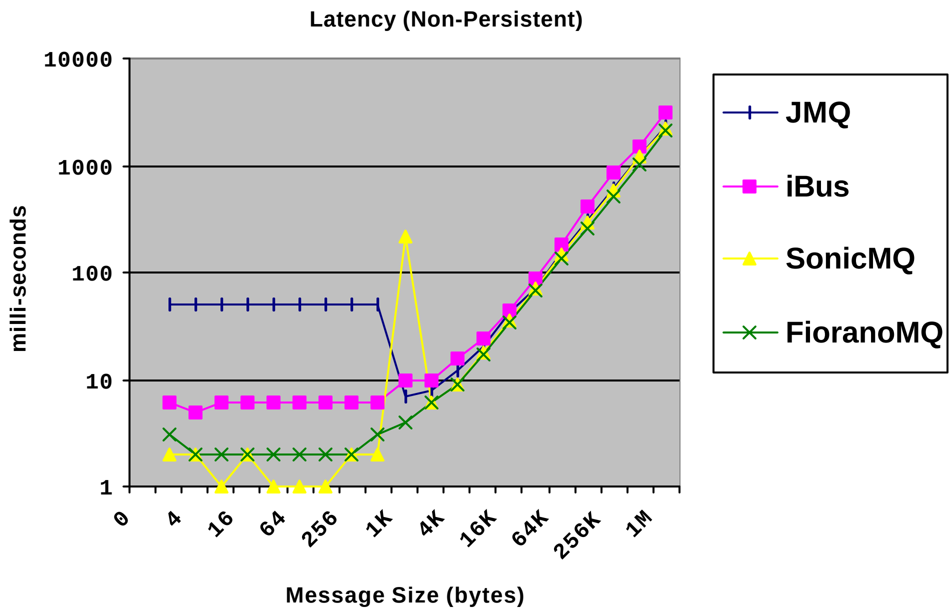
All GXOS Meta-objects have a unique name with a hierarchical structure and a URI of the form: *gndi://gxosroot/jpl/gem/virtual_California/run137/number_of_faults*

A.3.7 GMS Garnet or Grid Message Service

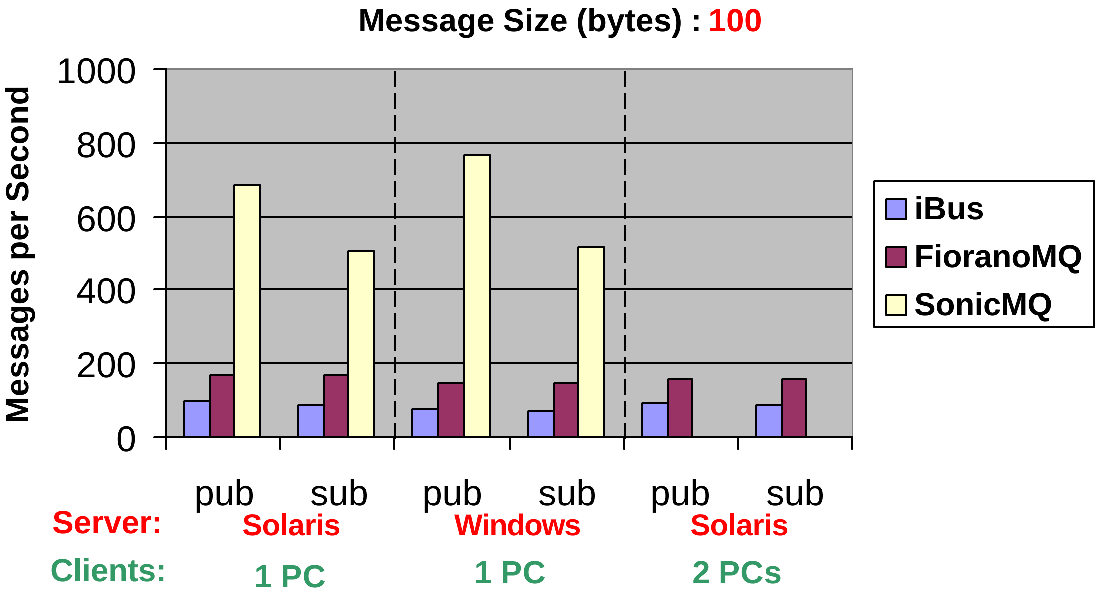
The publish/subscribe message based infrastructure is used to support ECCE and MyXoS. This supports XML based publication topics and subscription profiles as well as a sophisticated distributed server network supporting fault-tolerance and performance of message delivery. A research prototype is described in the June 2001 Syracuse PhD of Pallickara [18] while our initial "deployment" of ECCE uses JMS (Java Message service) as an interim solution. All messages are archived in an Oracle database. We expect to switch from JMS to a more powerful model as both our research and the work of the Grid Forum evolves. We are working through the Grid Computational Environment and Performance working groups to a consensus on a grid event service. In figures below, we note that JMS has much larger latencies (1 ms at best) than systems like MPI; JMS is high functionality and MPI high performance. These are both important but different optimizations. Note that 1 millisecond seems long by HPCC standards but is sufficient for collaboration systems and these measurements support our use of JMS as the core event system for synchronous collaboration.

Architecture of GMS: Grid Message Service





Non-Persistent/Non-Durabl
Number of Topic : 5
pub/sub per Topic : 10



A.3.8 ECCE Events

The events exchanged by the clients in ECCE are transmitted as time stamped messages by GMS and routed between clients using the publish/subscribe mechanism. The GXOS schema fully specifies ECCE event objects with all properties provided by the Schema.

A.3.9 GMSME or GMS Micro-edition

GMSME is the customization of GMS to small clients, which connect in MyXoS through "an adaptor" which is linked to a PDA or cell-phone class device via our HHMP (Hand Held Message Protocol). The adaptor (running on a conventional MyXoS server) performs functions such XML and SVG parsing, and rescaling of images. Typically the processing of any collaborative application (called a sharedlet in Garnet) is split between adaptor and client in a fashion that depends on client capabilities. A Windows CE Instant Messenger needs fewer services from the adaptor than the cell-phone IM interface. An adaptor looks like a GMS client to MyXoS and so this creates the illusion that GMS directly connects to small clients.

A.3.10 P2P Peer-to-Peer Systems.

P2P refers to a linkage of computers "at the edge" of the Internet. As shown in fig. 1, this can be achieved by routing through one or more servers. JXTA is a technology initiative by Sun [9] in this area. Systems like Napster are popular P2P environments. In the current Garnet, we use the same simple client-single server architectures used by the commercial collaboration systems and by our original Tango system. We are continuing to research this issue and develop approaches based on optimized routing of GMS messages based on a given server configurations and particular published topics and subscribed profiles [18]. Also dynamic instantiation of servers seems an important capability, which should be supported by MyXoS. For instance if one has each client at widely separated (in Internet land) locations, then a single server could be appropriate. If there are many clients in a given location, then MyXoS should generate dynamically a server at this location to implement optimal local P2P routing. Hardware multi-cast should of course be used if available.

A.3.11 MyXoS My eXtensible Web Operating System

This references the total environment including both the collaborative portal Garnet and the suite of administrative tools to manage the dynamic information infrastructure. At a low-level MyXoS is driven by scripted XML written in the W3C RDF syntax and referencing GXOS objects. MyXoS includes sophisticated search capabilities described below and an extremely interesting research challenge of defining how it brings referenced meta-objects into memory as requested by executing programs. This is termed the MyXoS execution model below. MyXoS will provide core systems services such as copy, create, grep (i.e. search), diff etc., familiar from UNIX and Windows.

A.3.12 RDF Resource Description Framework

RDF [19] is a W3C standard for metadata allowing any resource labeled by a URI to be given a value (which is either a literal or another resource) for a property. This can be used in MyXoS to specify or modify distributed tree fragments in a fashion similar to that used for distributed data sources in the Mozilla (Netscape 6) browser [20]. Each data source stores a fragment of tree; these are glued together by MyXoS as its distributed servers combine their information. Typical RDF uses in MyXoS are illustrated by the examples below.

*2)*Specify value for property in GXOS tree

```
    <rdf:description about="gndi://gxosroot/resourcename" ><gxos:property
    rdf:parseType="literal" >somevalue</gxos:property></rdf:description>
```

*3)*Specify profile by linking between GXOS tree elements

```
    <rdf:description about="gndi://gxosroot/sessionname"><gxos:userprofile
    rdf:resource="gndi://uri_of_user" gxos:customize="sessionspecificstuff"
    /></rdf:description>
```

*4)*Specify MyXoS copy command for meta-objects

```
    <rdf:description rdf:about="gndi://gxosroot/system/bin/cp"
    system:source="gxosobject1" system:destination="gxosobject2" gxos:execute="true"
    />
```

*5)*Specify alternative locations to find all FSU users

```
    <rdf:description aboutEachPrefix="gndi://gxosroot/users/fsu">
    <gxos:metaobjectlocation><rdf:alt>
    <rdf:li resource="http://main_fsuweblocation" />
    <rdf:li resource="http://backup_fsuweblocation" />
    </rdf:Alt></ gxos:metaobjectlocation></rdf:description>
```

Note that RDF is not an essential part of MyXoS and we can replace it by other XML based tools. Some people have expressed reservations about RDF because more powerful forms of knowledge representation may replace it. We will monitor the W3C and community activities and evolve our practice appropriately.

A.4 Asynchronous Collaboration and Object Management

ECCE and MyXoS provide a unified approach to sharing and managing objects; functions that are traditionally treated separately. For instance *Centra* is current leading e-learning collaboration system but it has weak management capabilities. *Blackboard* and *WebCT* are the leading learning management systems in Universities but neither is strong in collaborative capabilities. We combine the support of sharing and management because both require accurate metadata and we can achieve this with the same infrastructure MyXoS. Asynchronous collaboration is supported by uniform use of a single public/subscribe mechanism: currently JMS and to be extended to GMS. In principle all forms of asynchronous collaboration would be included in GMS by wrapping if necessary "foreign objects" like email as a GMS event. We will do this as needed but it is not realistic for us to redefine ways of working that are already adequate. ECCE does use the public domain Jabber instant messenger [21] and has modified this to interface with GMS. Synchronous collaboration is integrated with the asynchronous system by using the same publish/subscribe mechanism GMS for it.

A.4.1 Registration of Meta-Objects

MyXoS maintains queues (topics) to which aggregates and meta-objects can post their location. These messages are similar in function to those used in *Jini* and are automatically purged when their posted validity expires. These messages contain one or more RDF statements identifying the location of the meta-objects with certain value or range of URI (GNDI names). Currently this can be most easily specified using the *about* or *aboutEachPrefix* attribute in *rdf:description* tag (see sec. A3.12) but we expect to generalize this rather simple syntax. These locations contain either the meta-objects or further RDF statements giving you directions. In this fashion MyXoS uses distributed engines subscribing to the registration topics to build up indices to map GNDI name into meta-object location. The meta-object is either the desired object or uses the GXOS Object realization Schema to specify access to the original object.

A.4.2 Efficient Handling of Objects

We need to combine the high flexibility and functionality of distributed objects with the performance associated with traditional analysis systems using simple flat files with customized formats. We must satisfy the requirements of security, collaboration and distributed dynamic objects. The most important general strategy is our use of small meta-objects, which contain the essential information for implementing MyXoS services. The original object is only accessed when necessary. The information stored in the meta-object is application dependent and requires careful design of the GXOS extension for this object type. This will involve structured data, which will be aggregated, and the ECCE Gallimaufry, which includes everything else such as reports, notes and presentations. There are fewer in number of this type of data but they are much more heterogeneous in content and scattered over file systems around the world. Basic events are expected to be consolidated in a few places and so amenable to special processing. This careful aggregation will allow MyXoS to be implemented efficiently and deal with a realistic (billions not quadrillions) number of meta-objects.

The requirement of all objects to have metadata has important implications as an enabling technology for the management of science research as it grows in size and complexity. We believe such discipline (enforced meta-data) will be very important in ensuring success as collaborations grow in size.

A.5 MyXoS Execution Model

We are currently researching different ways of reading into memory the XML meta-objects as needed by programs running under MyXoS. SAX and DOM XML parsers are not efficient for tens of millions of XML instances at a time. Converting XML schema into Java data structures is possible [22] but efficiency requires this be combined with "lazy" parsing so that we expand GXOS trees only as needed to refine our access. Remember the use of multi-resolution aggregates "stopping" at a certain level in GXOS tree is absolutely essential for efficient systems. We see this as a particularly challenging problem that has important programming style implications as we look at new computing paradigms where data structures are defined in XML and not directly as C++ or Java classes.

Elsewhere we intend to look at areas like "Parallel Computing in MyXoS" with intelligent XML based data structures interpreted by Java agents enabling both more powerful decomposition (of the XML structures and algorithm expressed either in Java or some more powerful version of RDF) strategies and the use of distributed dynamic resources for parallel execution. In this model we will as now produce MPI based SPMD codes but with a very different way of specifying the problem. We believe the research proposed here would be synergistic with such other applications of emerging Web operating environments like MyXoS. Another general capability needed by all these problems is "packed or binary XML" which can most efficiently represent XML structured for optimal parsing.

Bibliography

1) Portals and Frameworks for Web-based Education and Computational Science, http://www.new-npac.org/users/fox/documents/pajavaapril00/

2) General Review of Portals by G. Fox, http://aspen.csit.fsu.edu/collabtools/CollabReviewfeb25-01.html

3) Gateway Computational Portal, http://www.gatewayportal.org

4) NEES Earthquake Engineering Grid, http://www.neesgrid.org/

5) Particle Physics Data Grid, http://www.cacr.caltech.edu/ppdg/

6) GriPhyN Project Site, http://www.griphyn.org/

7) Ninja Project, http://ninja.cs.berkeley.edu/

8) Semantic Web, http://www.w3.org/2001/sw/ and article in May 2001 Scientific American by Tim Berners-Lee, James Hendler and Ora Lassila at http://www.sciam.com/2001/0501issue/0501berners-lee.html

9) Sun Microsystems's JXTA, http://www.jxta.org and http://conferences.oreilly.com/p2p/p2p_brochure.pdf

10) The Tango System, http://www.new-npac.org/users/fox/documents/generalportalmay00/erdcportal.html

11) Virtual Network Computing System, http://www.uk.research.att.com/vnc

12) X3D http://www.web3d.org/x3d.html

13) Access Grid, http://www.mcs.anl.gov/fl/accessgrid/

14) HearMe, http://www.hearme.com

15) Instructional Management Systems (IMS), http://www.imsproject.org

16) Advanced Distributed Learning Initiative, http://www.adlnet.org

17) Grid Forum Computing Environment Working Group, http://www.computingportals.org/cbp.html

18) Grid Message (Event) System, http://aspen.csit.fsu.edu/users/shrideep/mspaces/. This will be June 2001 Syracuse PhD of Shrideep Pallickara.

19) Resource Description Framework (RDF), http://www.w3.org/TR/REC-rdf-syntax/

20) RDF implementation in the Netscape Browser, http://www.mozilla.org/rdf/doc/

21) Jabber Instant Messenger, http://www.jabber.org

22) Castor: XML Schema to Java Data Structure Converter, http://castor.exolab.org/sourcegen.html

23) The Grid Forum http://www.gridforum.org

24) The open Peer to Peer P2P organization http://www.openp2p.com

25) XSIL from Roy Williams at Caltech, http://www.cacr.caltech.edu/SDA/xsil/index.html

26) WebCT Commercial Education Portal http://www.webct.com

27) Blackboard Commercial Education Portal http://www.blackboard.com

28) Akarsu, E., Fox, G., Furmanski, W., and Haupt. T., ``WebFlow-High-level Programming Environment and Visual Authoring Toolkit for High Performance Distributed Computing," in *Proceedings of Supercomputing '98,* August 15, 1998.

29) Erol Akarsu, Geoffrey Fox, Tomasz Haupt, Alexey, Kalinichenko, Kang-Seok Kim, Praveen Sheethaalnath, and Choon-Han Youn, "Using Gateway System to Provide a Desktop Access to High Performance Computational Resources", Proceedings of *HPDC-8 Conference, Redondo Beach Ca., Aug 3-6, 1999, IEEE Press.*

30) Tomasz Haupt, Erol Akarsu, Geoffrey C. Fox, "Landscape Management System: A WebFlow Application", ERDC Technical Report August 1999(updated), http://www.new-npac.org/users/fox/documents/LMSERDCC425/C425LMSrep2.html

31) HDF5 scientific file format http://hdf.ncsa.uiuc.edu/HDF5/

32) ICE Interdisciplinary framework and eXtensible Data model and Format (XDMF), http://www.arl.hpc.mil/SciVis/dice/ and http://www.arl.hpc.mil/PET/training/SEM77.html

33) Mehmet Sen, Syracuse PhD Thesis November 2000, "Distributed Open Asynchronous Information Access Environment" at http://aspen.csit.fsu.edu/collabtools/senthesisdraft.html

34) XML Query Working Group http://www.w3.org/XML/Query.html

35) URI Universal Resource Identifier, IETF RFC 2396 August 1998, Tim Berners-Lee, R. Fielding, L. Masinter, http://www.ietf.org/rfc/rfc2396.txt

36) Java Message Service http://java.sun.com/products/jms/index.html