ENABLING INTERACTIVE AND COLLABORATIVE COMPUTATIONAL SCIENCE ON THE GRID

Preeti Mehra and Manish Parashar
The Applied Software Systems Laboratory,
ECE/CAIP, Rutgers, The State University of New Jersey
E-mail: {mehra, parashar}@caip.rutgers.edu

ABSTRACT

A *collaboratory* is a virtual environment where geographically scattered scientists and engineers come together virtually to solve complex problems. In this paper we investigate the requirements of a scientific computational collaboratory by identifying the characteristics of the interactions in such a system. We then identify the mechanisms and tools that can enable realistic collaborative work in a scientific computational environment, and describe the incorporation of these tools into the Discover computational collaboratory. Specifically, we present the design and implementation of a workspace management approach, termed as Interaction Streams, for dynamic organization, representation and analysis of personal and shared workspaces in Discover.

1. INTRODUCTION

A *collaboratory* has been defined as a group of people working together from diverse physical locations on a common task using a shared workspace. A *computational collaboratory* is a virtual environment where scientists and researchers work together to solve complex interdisciplinary problems, despite geographic and organizational boundaries. These systems provide uniform, pervasive (and collaborative) access to computational resources, services, applications and/or data, and can expand the resources available to researchers, enable multidisciplinary collaborations and problem solving, accelerate the dissemination of knowledge, and increase the efficiency of research. The emergence of Grids-based computational collaboratories and the potential for seamless aggregation, integration and interactions has made it possible for scientists and engineers to conceive a new generation of realistic, scientific and engineering simulations of complex physical phenomena.

In this paper we first investigate the collaboration requirements for a computational collaboratory. These requirements are based on the nature of interactions between users and their interactions with applications, services and data. We then present the design and implementation of a portal for collaborative science and engineering on the Grid. The portal enables geographically distributed scientists and engineers to collaboratively deploy, access, monitor, interact with and steer Grid applications and services by providing collaboration tools. Finally we describe a workspace management and organization tool for private/shared workspaces within this portal. The design, implementation and experimental evaluation of this tool are presented.

The portal is a part of the Discover computational collaboratory [3][9]. Discover is a virtual, interactive computational collaboratory that enables geographically distributed scientists and engineers to collaboratively access, monitor and control high performance parallel/distributed applications on the Grid. Its primary goal is to bring Grid applications to the scientists'/engineers' desktop, enabling them to collaboratively access, interrogate, interact with and steer these applications using pervasive portals.

2. BACKGROUND and Related Work

The aim of a collaborative computational problem solving environment is to provide scientists with a set of mechanisms and tools for collecting, storing, manipulating, accessing and sharing information using meaningful annotation tools supporting their views and ideas. When accessing such an environment, a scientist should be able to reliably and transparently initialize, deploy, access, monitor and steer applications while collaborating with other scientists. Central to the design of such a collaborative scientific problem solving environment is understanding the types of interactions taking place between the participating entities and, identifying their corresponding enabling tools and mechanisms. Global scientific investigation involves seamless interactions between all the participating entities – i.e. users (scientists and engineers), data sources, applications, and resources. User to user(s) interactions can be defined as the sharing of any human-understandable information, being textual, visual or audible, among a managed group of interested parties. Applications interact with resources and data centers when they need to allocate resources, to fetch data to perform a certain computation or to store the results from a computation. Users interact with applications, data and resources to monitor, control and steer the application and/or resources and to explore the results produced. Based on these interactions we explore the set of tools and services for enabling collaborative computational problem solving.

Group management allows users to create or join groups by interest. Every group needs to be maintained by administrators who have special privileges for altering the group configuration, such as the credentials and capabilities required for joining a group, privileges of each group member and total number of members allowed.

Monitoring and steering allows users to (synchronously or asynchronously) access, monitor and control application objects (as well as services) in a secure, controlled and consistent way. **Sharing tools** associated with the results from queries allow users to share specific results with a group of users. **Data manipulation tools** allow users to render the results obtained to enhance their analysis and interpretation.

Locking/leasing mechanisms are necessary to maintain consistency when an application is concurrently accessed by multiple users but can only support one control channel. A lease is a lock (or other privilege) that is valid for a fixed duration. Locks/leases are only granted to users that have appropriate privileges and capabilities.

Collaboration tools such as **Chat and Whiteboard** provide users with a means for collaborating while analyzing the results. The whiteboard also provides a mechanism for **annotation**, allowing users to mark, draw, write or point to information on rendered application results.

Logging and Management tools are necessary for maintaining a history of the sequence of events that occurred in the system, including user requests, application responses or status messages and user-user collaborations. These tools are critical for allowing scientists to a posteriori analyze and interpret results of interactions and collaborations.

The overall goal of the portal presented in this paper is to integrate the set of tools outlined above to effectively support computational collaboratories. A key focus is the design of an effective workspace management tool to provide a navigable record of all user-user and user-application interactions and collaborations. General collaborative environments such as Groove [8] or NetMeeting [12] provide users with tools for sharing files, text/voice/video messages and applications. However, this is done in a generic manner without taking the specific requirements of an application into account, and provides a set of standardized tools to the collaborating users. Recent scientific collaboratory efforts include the Upper Atmospheric Research Collaboratory (UARC) [14] and Astrophysics Simulation Portal (ASC Portal) [13]. UARC provides an environment for space physicists around the globe to collaborate and share information collected by upper atmospheric instruments. ASC Portal, built on Cactus [2] and the Globus toolkit, is a framework that allows scientists to deploy and monitor simulations on the computational Grid. While these frameworks do address collaboration, their tools are highly customized to the application domain.

Lifestreams [4-7] is a metaphor for the organization and storage of electronic documents is an alternative to the traditional desktop paradigm and its associated folder and file based storage system. It consists of a time-ordered stream of documents, which may or may not be related, and which may have originated from a wide array of sources. The DISCIPLE collaborative environment provides a tool called Event Streams [1] for organization of the shared workspaces by session logging and retrievals. We feel that the concepts embodied by the Lifestreams interface and Event Streams can be adapted to form a tool for workspace management and review of the collaborative and private sessions that are the basis of the Discover project.

3. THE DISCOVER COMPUTATIONAL COLLABORATORY

Discover, a web-based computational collaboratory, is designed as a three-tier architecture comprised of pervasive (detachable) client portals at the front end, a peer to peer network of interaction servers in the middle and a control network of sensors, actuators and interaction agents superimposed on the application at the backend. The user can invoke a client from a web browser by authenticating to the servers. The client when connected to the server receives a list of active running applications to which it has some access privileges, and may be able to launch new applications. The middle tier consists on server peer that extend commodity web servers with interaction and collaboration capabilities. This server peer is responsible for translating and forwarding requests and commands from the clients to the control network associated with the applications at the backend. Some of the other functionalities built into server peers are session management, locking, concurrency control, and security and authentication. A detailed description of the Discover architecture can be found in [9-11]. In this paper we focus the design of the Discover collaborative portal.

Prior to joining the collaboratory, each user has to register with the system and is assigned a username and password as well as a set of privileges per application of interest. Each user joins an application group by authenticating and presenting his/her privileges to the authentication handler. Changing privileges cannot be done dynamically and has to be initiated by the coordinator(s) of the group(s) to which the users belongs. In the Discover framework, a default group (the application group) is created when an application connects to the framework and all users accessing the application become members of this group. All requests, commands and responses issued to/by the application are broadcast to this group to ensure a consistent view of the application. Users can also create topic-based virtual rooms in which they can exchange information.

3.1 The DISCOVER Collaborative Portal

In Discover, users/scientists have access to remote distributed application objects through a virtual desktop interface, which essentially provides a monitoring and control window to executing applications. The desktop is divided into global and a local workspaces. The components of the desktop can be moved between the global and local desktops. The global desktop is a replicated shared workspace for the user's collaboration group. It is composed of panels displaying the application object list, application responses, global and local update messages, and the list of users collaborating in the default application group. The object list contains the objects exported by the application and its associated sensors, actuators and interaction methods for monitoring and steering them. Application objects are dynamically exported to each user, allowing runtime object addition or deletion.

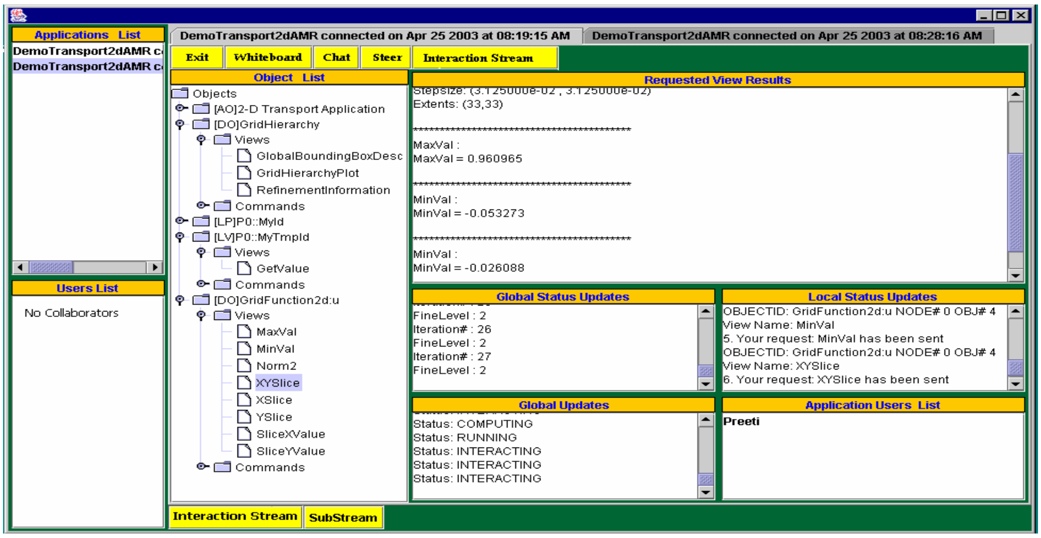

Figure 1. A snapshot of the Discover portal.

The communication aspect of the collaboration is enabled through chat text messages and a whiteboard that allows users to annotate visual data in a shared and synchronized manner. Every graphical view (e.g. 2D or 3D plots) can be annotated using marking tools (drawing, writing) customized (by style or color) for each user. In the collaboration mode every user's work is distinguished by the means of different color schemes and telepointers. Locking tools provided ensure the consistency of the global workspaces and restrict issuing of steering commands to the user owning the lock. An innovative component of the portal is *Interaction Streams*, which provides workspace management in the collaborative environment. It not only is an interaction-cataloguing tool for a user's workspace, but is also a search tool and visual representation tool for extracted information from the logs. Figure 1 shows a screenshot of the Java based Discover portal running on a desktop computer and interacting with two applications connected to the Discover server.

4. INTERACTION STREAMS

Interaction Streams in Discover maintain navigable records of all user-user and user-applications interactions and collaborations. Each user has a personal Interaction Stream consisting of events occurring in their local and shared workspaces. Events can be application/service or user initiated, and are further divided into subcategories based on the information they contain. Stream filters are provided to organize and present the information from these streams. Visual representations of these streams consist of a stack of time ordered documents depicting the events. Features such as borders, titles and icons complement the documents based on the keys to ensure rapid accessibility and categorization. Intersections on Interaction Streams define interactions and collaborations between groups of users in space and time. These intersections are monitored by agents and take into account user credentials and privileges in the system. Intersections hold answers to questions such as "did a group of users ever collaborate", "on which applications or in what context did a group of users collaborate", or "which interactions were done by a particular group of users and on which applications". Interaction Streams are divided into two categories, User Streams and Application Streams.

4.1 User Interaction Stream

A user's interaction stream is a comprehensive log of the interactions occurring in his/her private/public workspace throughout his/her usage of the system. Security constraints are enforced to prevent unauthorized accesses to these streams and users are allowed to only access the streams that they own. Every record in the stream is a document that contains five basic elements that entirely describe an interaction. The basic elements are Application Identifier, Interaction Identifier, Time of Interaction, and Description of Interaction. Time is depicted as a combination of the date and time the interaction was received at the portal. A sequential arrival of the interactions at the user enforced by queues maintained at the peer servers guarantee a unique timestamp. Different interactions present in the system include Client Login, Client Disconnect, View Response, Steer, Global and Local Updates, Chat and Whiteboard Interactions. These streams are implemented as tables in a relational database. The stream handler at the server manages the insertions of these interactions as rows in the tables. In order to reduce the overhead introduced by database access, the interactions are buffered at the server and multi-valued insertions are made.

4.1.1 Graphical Interface for the User Interaction Stream

The graphical interface for the user stream is composed of a sequence of time-ordered documents arranged from present to the past. The date annotated scroll bar provided at the bottom of the user stream interface provides the capability of flipping through this stack of the documents. A record can be brought to focus by selecting it from the stack and results in the roll over of the document to the front and a refresh of the stack with the appropriate information. Every record holds information about the state of the application or the action taken by the user.

Records are divided into regions to display the elements of the interaction. These regions are characterized by the use of colored borders and icons associated with them. Icons are also used to represent different kinds of interactions occurring in the system. These records present concise information of the interaction enabling rapid access to this information. Only the main characteristics of an interaction are displayed in the Description region of the record in the interface and the overview button presents the user with a complete view of the record in a separate window. Figure 2 shows the graphical interface for a user stream and the complete description of the record in focus. Memory is a constraint as the number of records can be large. An approach similar to paging is used here, where the user is presented with an illusion of a continuous stream of all the records but only a working set of active record is maintained. New records are replaced in the set when the user slides away from the active set using the scroll bar.

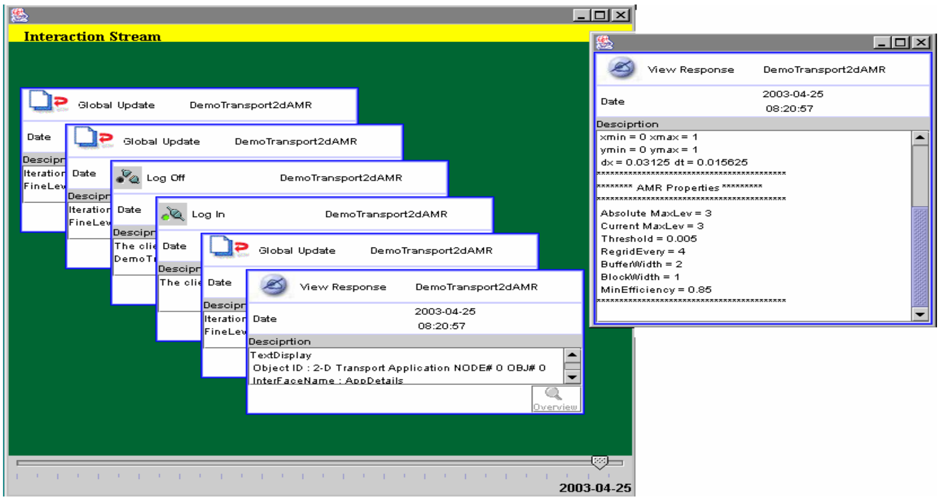

Figure 2. Graphical interface for Interaction Streams

4.1.2 Graphical Interface for Sub-streams

The size of the interaction stream grows as interactions occur during the lifetime of the user's workspace. Searching through these records to find a record or subset of records of interest is very inefficient in terms of time and effort. To address this requirement, a Sub-stream Tool has been design and added to the system. Figure 3 below shows the graphical interface for the tool. The tool performs two operations – the construction of a sub-stream from the user's personal stream and from the intersection of the collaborators streams. The view consists of 5 elements:

- Application List: This is the list of the applications that the user is registered with in the Discover system. This list is obtained from the Discover server.
- Collaborators List: This list consists of other users who are registered to the same applications as the user and hence are termed as the collaborators.
- Time From: This allows the user to select the lower limit of the time period of interest.
- Time To: This allows the user to select the upper time limit of the time period of interest.
- Event List: This list consists of the kinds of events of interest. As described above, events are divided into eight categories. All categories are contained in this list.

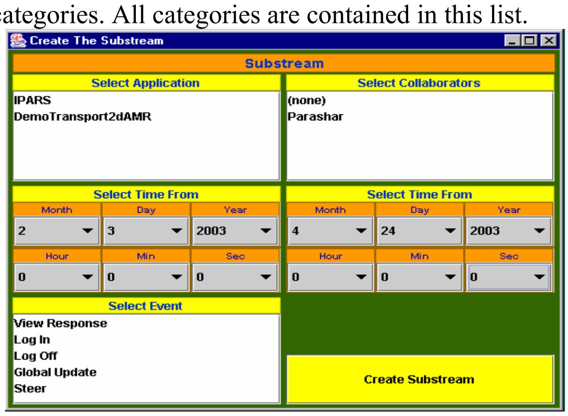

Figure 3. Graphical interface for Sub-streams

Non-editable lists are generated by accessing the information in the Discover server database resources, preventing unauthorized access into another user's stream. Substream creation is probably the most important feature provided in this interaction tool. Selecting a selection or all criteria, one can now obtain a stream of filtered documents. Multiple values can be selected from any of the lists to construct the filtering criterion for the sub-stream. The Create Sub-stream Button when clicked gathers the selected conditions and creates a union of all these conditions. As the streams are created on the fly from the streams stored in the system, the stack created is dynamic and no permanent storage is assigned to it. All the documents satisfying these conditions now form the new sub-stream and are displayed in a graphical interface similar in structure to the user stream graphical interface.

The selection of values from the list of collaborators causes an intersection between the user's stream and the streams of the selected collaborators. The sub-stream thus created contains interactions that occurred during the collaboration of selected collaborators and contain same values of the elements Time and Application Identifiers. Information presented in the shared workspace is customized based on the privilege levels of the collaborators. For example, a user's privilege level may not allow the user to view responses to request by collaborator. Due to the synchronous nature of the Discover system, users need to be in collaboration mode to have a non-empty intersection of records in the streams.

4.2 Application Interaction Streams

An Application Stream contains the complete log interactions during an application's life cycle, such as the status of the objects at regular time intervals, the users registered, the time the users logged in to interact with the application, the monitoring requests and steering commands, etc. This stream enhances the security of the system as it provides a powerful medium to audit the application by the administrator of the application. The owner of the application, who registers application to the Discover server, can only access its application stream. The graphical interface of the stream is similar in structure to that of the User Interaction Stream.

5. EXPERIMENTAL EVALUATION

5.1 Experiment 1

This experiment aims at quantifying the overheads introduced by the addition of the logging interactions at the Discover server end. The experiment consisted of calculating the time required to empty the buffer into the database. The overheads measured in this experiment are plotted in Figure 4. The peaks in the graph represent the time taken to perform the group logging from the buffer. There are intervals of time where no record is logged to the database. An overhead of 6.05ms on an average is introduced per interaction at the server.

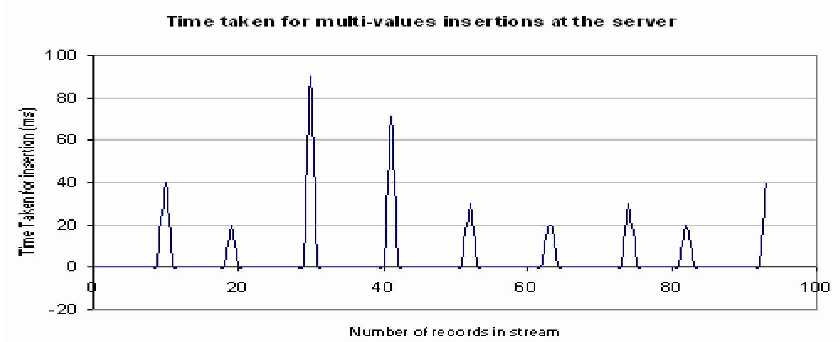

Figure 4. Graph showing time overheads for logging Interaction Streams

5.2 Experiment 2

This experiment measures the memory used for the storage of records in the interaction streams. The graph in Figure 5 shows a linear increase in the memory used with the increase in the number of records indicating a almost constant memory requirement per record. On an average a record consumes 220 bytes.

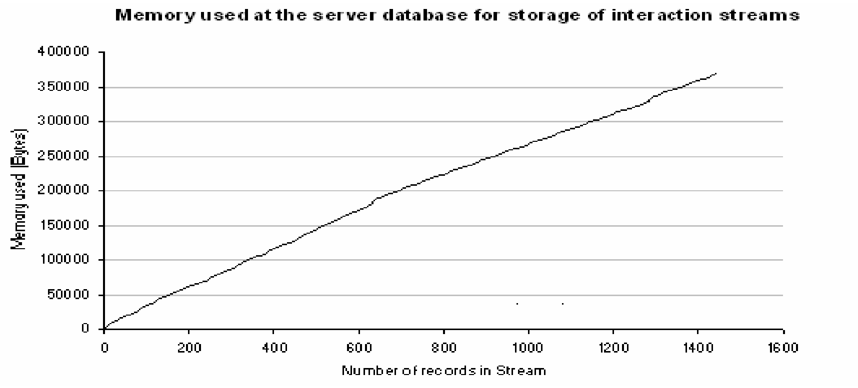

Figure 5. Graph showing memory usage by Interaction Streams

5.3 Experiment 3

This experiment is used to estimate the time overheads in the representation of the Interaction Streams. The experiment consisted of calculating the time required to load the active set of records from a user interaction stream that satisfy a user defined criteria. The access times are plotted against the total number on records contained in the stream. The graph in Figure 6 shows that the access times of these records are between 10 ms and 30 ms (depending on the sizes of the records retrieved) and do not increase with the size of the stream as only a subset of the information is retrieved at any point.

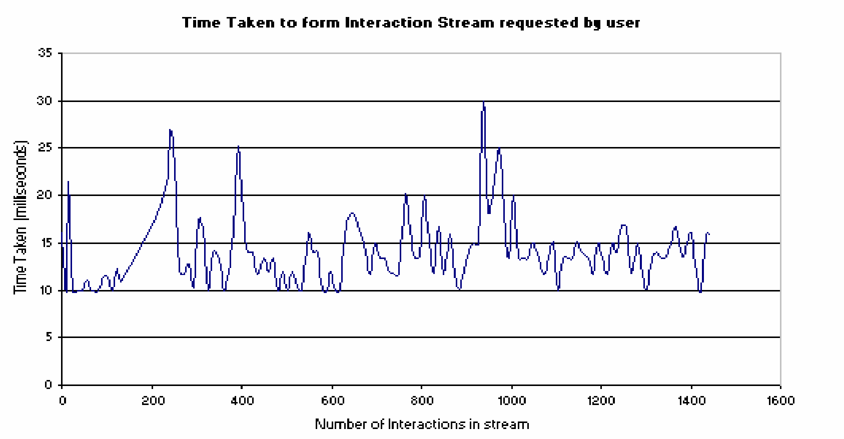

Figure 6. Graph showing time required to extract records from Interaction Streams

6. SUMMARY

In this paper we first identified the collaboration requirements for scientific problem solving and the required tools and mechanisms for enabling this collaboration. We then presented the Discover computational collaboratory for application interaction and steering. Discover provides collaborative tools that support these requirements, and allow geographically distributed scientists and engineers to collaboratively access, monitor and control distributed simulations. The concept of Interaction Streams was introduced as a means for maintaining comprehensive logs of private and collaborative sessions. The design and implementation of this tool for workspace organization and analysis winthin the Discover computational collaboratory was described. An experimental evaluation of the tool and its overheads was presented.

ACKNOWLEDGEMENT

Support for this work was provided by the NSF via grants numbers ACI 9984357 (CAREERS), EIA 0103674 (NGS) and EIA-0120934 (ITR), DOE ASCI/ASAP (Caltech) via grant numbers PC295251 and 1052856.

REFERENCES

[1] Bianco, M., 2000. An Interface for the Visualization and Manipulation of Asynchronous Collaborative Work within the DISCIPLE System, Department of Electrical and Computer Engineering, Rutgers University, USA.
[2] Cactus Computational Collaboratory, http://www.cactuscode.org.
[3] Discover Computational Collaboratory, http://www.discoverportal.org.
[4] Fertig, S. et al, 1996. Lifestreams: An Alternative to the Desktop Metaphor, In *ACM SIGCHI Conference on Human Factors in Computing Systems Conference Companion (CHI'96)*, pp. 410 - 411, ACM Press.
[5] Freeman, E. et al, 1995. Lifestreams: Organizing your electronic life, in *AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval.*
[6] Freeman, E. et al, 1996. Lifestreams: A Storage Model for Personal Data, *ACM SIGMOD Bulletin*, USA.
[7] Gelernter, D., 1998. ScopeWare - Information Management Systems, Mirror Worlds Technologies. http://www.mirrorworlds.com/horizons/index.html.
[8] Groove Networks. http://www.groove.net.
[9] Mann, V. et al, 2001. Discover: An Environment for Web-based Interaction and Steering of High-Performance Scientific Applications, Concurrency and Computation: Practice and Experience, John Wiley and Sons, Vol. 13, Issue 8-9, pp 737 - 754.
[10] Mann, V. et al., 2002. Engineering an Interoperable Computational Collaboratory on the Grid, *Concurrency and Computation: Practice and Experience, Special Issue on Grid Computing Environments*, Editors: G. Fox, D. Gannon, M. Thomas, John Wiley and Sons, Vol 14, Issue 13-15, pp. 1569-1593.
[11] Muralidhar, R. et al, 2001. A Distributed Object Infrastructure for Interaction and Steering, *Proceedings of the 7th International Euro-Par Conference (Euro-Par 2001)*, Lecture Notes in Computer Science, Editors: R. Sakellariou, J. Keane, J. Gurd and L. Freeman, Springer-Verlag, Manchester, UK, Vol. 2150, pp 67 - 74.
[12] NetMeeting http://www.microsoft.com/netmeeting.
[13] Russell, M. et al, 2001. The Astrophysics Simulation Collaboratory: A Science Portal Enabling Community Software Development, Proceedings of *10th IEEE International Symposium on High Performance Distributed Computing*, San Francisco, CA, USA, pp 207-215.
[14] Subramanian, S. et al, 1999. The UARC web-based collaboratory: Software architecture and experience, *IEEE Internet Computing*, Vol. 3, No. 2, pp.46-54.