



DISCOVER: A VIRTUAL INTERACTIVE COMPUTATIONAL COLLABORATORY

BY DHILLON, S. KAUR, V. MANN, V. MATOSSIAN, R. MURALIDHAR, A. SWAMINATHAN, AND M. PARASHAR

The Applied Software Systems Laboratory (TASSL), CAIP/ECE
<http://www.caip.rutgers.edu/TASSL/>

DISCOVER (*Distributed Interactive Steering and Collaborative Visualization EnviRonment*) is a virtual, interactive computational collaboratory that enables geographically distributed scientists and engineers to collaboratively monitor, and control (new and existing) high performance parallel/distributed applications. Its primary goal is to bring large (remote) distributed simulations to the scientists/engineer by providing collaborative web-based portals for interrogation, interaction and steering. DISCOVER has a 3-tier architecture (see Figure 1) composed of detachable client portals at the front-end, a network of interaction servers in the middle, and a control network of sensors, actuators, and interaction agents superimposed on the application at the back-end. Clients can connect to a server at any time using a browser to receive information about active applications. Furthermore, they can form or join collaboration groups and can (collaboratively) interact with one or more applications based on their capabilities. The interaction servers build on *serverlet* technology and enable clients to connect to, and collaboratively interact with registered applications using a conventional browser. The application control network enables sensors and actuators to be encapsulated within, and directly deployed with the computational objects. Interaction agents resident at each computational node register the interaction objects and export their interaction interfaces. These agents coordinate interactions with distributed and dynamic computational objects. The application interaction gateway manages the overall interaction through the control network of interaction agents and objects. It uses the Java Native Interface (JNI) to create Java proxy objects that mirror the computational objects and allow them to be directly accessed by the interaction web-server. Security and authentica-

tion services are provided using customizable access control lists built on the SSL-based secure server.

The DISCOVER interaction model is application initiated, i.e. the application registers with the server exporting an interaction interface composed of "views" and "commands" for different application objects using high-level interfaces. Views encapsulate sensors and provide information about application and application objects, while commands encapsulate actuators and process steering requests. Some or all of these views/commands may be collaboratively accessed by groups of client based on the client's capabilities. DISCOVER is currently operational and being used to provide interaction capabilities to a number of scientific and engineering applications, including oil reservoir simulations, computational fluid dynamics and numerical relativity.

DISCOVER INTERACTION AND COLLABORATION SERVERS

The DISCOVER interaction/collaboration server builds on a traditional web server and extends its functionality to handle real-time information, and to serve client requests and application connections. Extension is achieved using Java servlets (server side Java programs) to add an extendible set of specialized services. Each server consists of a number of handler servlets running to provide different interaction and collaboration services. Clients connect to the server using standard HTTP communication using a series of HTTP GET and POST requests. At the other end, application-to-server communication is achieved either using standard distributed object protocols like CORBA and Java RMI, or a more optimized, custom protocol using TCP sockets. The core service handlers provided by each server include, the Master Handler, Collaboration Handler, Command Handler, Security/Authentication Handler and a Daemon Servlet that listens for application connections. In addition to these core handlers, there can be a number of handlers providing auxiliary services such as session archival, database handling, visualization, request redirection, and remote application proxy invocations (using CORBA).

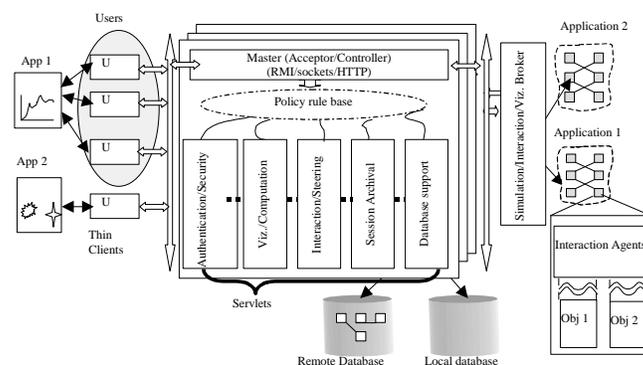


Figure 1 – Architectural Schematic of the DISCOVER Computational Collaboratory

DISCOVER (CONTINUED)

APPLICATION CONTROL NETWORK FOR INTERACTION AND STEERING

The control network of interaction agents has a hierarchical cellular structure and partitions the processing nodes into a number of interaction cells. The network is composed of (1) Discover Agents on each node, (2) Base Stations for each interaction cell and (3) an Interaction Gateway that connects to the interaction server and provides a proxy to the entire application. The number of nodes per interaction cell is programmable. The cellular control network is automatically configured at run-time using an underlying messaging environment (e.g. MPI) and the available number of processors.

Discover Agents present on each node maintain run-time references to all registered interaction objects on that node. Since object references can change dynamically during program execution, the interaction agents ensure that object references are valid and refer to consis-

create Java mirrors of each registered interaction object. Thus, interaction Web Servers can directly access the application's interaction objects (and thus the computational objects) using standard distributed object interfaces like Java RMI.

THE COLLABORATIVE INTERACTION AND STEERING PORTAL

The DISCOVER collaborative computational portal is a working environment for scientists/engineers, that seamlessly integrates access to DISCOVER services, and empowers them with an anytime/anywhere capability of collaboratively (and securely) monitoring and controlling applications, independent of platform architecture or geographic location. Figure 2 shows a screen dump of the current portal. The base portal, presented to the user after authentication and application selection, is a control panel. The control panel is designed to be lightweight as all clients irrespective of their capabilities must be able to download it. Once the client has the control panel s/he can launch any desired service such as view interroga-

tion, interaction, collaboration, or application/session archival access. The application control panel consists of: (1) a list of interaction objects and their exported interaction capabilities (views and/or commands), (2) an information pane that displays global updates from the selected application, and (3) a status bar that displays the current mode of the application (computing, interacting) and the status of command/view request. The list of interaction objects is customized to match the client's access privileges. Chat and white-

"Clients can connect to a server at any time using a browser to receive information about active applications."

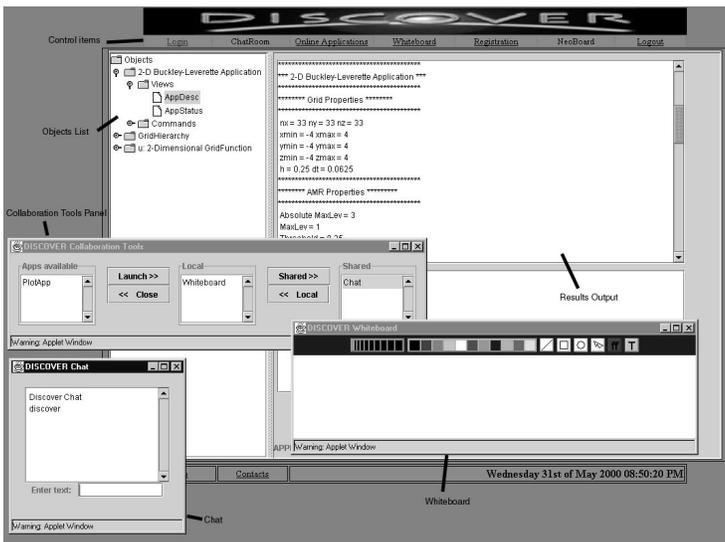


Figure 2 – The DISCOVER Collaborative Interaction/Steering Portal

tent data. Discover Agents, Base Stations and the Gateway each maintain registries of interaction objects registered in their respective domains (node, cell, entire application, respectively). The Gateway is additionally responsible for interfacing with the interaction server, delegating interaction requests to the appropriate interaction agents (Discover Agents and/or Base Stations), and collecting their responses. In the case of distributed objects, the Gateway also performs a gather operation for collating the responses arriving from the corresponding nodes. Furthermore, the Gateway uses JNI to

board tools enable collaboration. View requests generate separate panes using the corresponding view plug-in. A separate application registration page is provided to allow super-users

to register applications, add users and modify user capabilities.

MORE INFORMATION

Further information about the DISCOVER collaborative and its applications can be found at www.caip.rutgers.edu/TASSL/Projects/DISCOVER.