WebHLA - An Interactive Multiplayer Environment for

High Performance Distributed Modeling and Simulation

Geoffrey C. Fox, Ph. D., Wojtek Furmanski, Ph. D.,

Subhash Nair, Hasan T. Ozdemir, Zeynep Odcikin Ozdemir, Tom A. Pulikal

Northeast Parallel Architectures Center, Syracuse University

111 College Place, Syracuse University, Syracuse NY 13244-4100

{gcf, furm, subhash, timucin, zeynep, tapulika} @ npac.syr.edu

Keywords:

Gaming, Interactive Simulation, Military, Personal Computers, Standards

ABSTRACT

The process of integrating DoD Modeling and Simulation paradigms around the new HLA/RTI standards proceeds in parallel with the onset of new Object Web standards for distributed objects and componentware, emergent at the crossroads of CORBA, DCOM, Java, and XML based distributed object technologies. We describe here our WebHLA approach which integrates both trends by offering Object Web based implementation of the HLA framework. WebHLA follows a three-tier architecture including: a) Web / Commodity based interactive simulation authoring and runtime front-ends given by Java applets or DirectX multiplayers; b) software bus in the middleware, given by Object Web RTI i.e. our implementation of DMSO RTI 1.3 as a Java CORBA service managed by JWORB (Java Web Object Request Broker); c) domain specific backend simulation modules, including advanced high performance DoD simulation kernels such as ModSAF or SPEEDES. We describe here the overall design of WebHLA, we report on components prototyped so far and we summarize the status of the ongoing development of WebHLA applications.

1. INTRODUCTION

We present here early results of our work on Web / Commodity based High Performance Modeling and Simulation, conducted as part of the academic branch of the Forces Modeling and Simulation (FMS) domain within the DoD HPC Modernization Program. Our approach explores synergies between ongoing and rapid technology evolution processes such as: a) transition of the DoD M&S standards from DIS to HLA; b) extension of Web technologies from passive information dissemination to interactive distributed object computing offered by CORBA, Java, COM and W3C WOM; and c) transition of HPCC systems from custom (such as dedicated MPPs) to commodity base (such as NT clusters).

One common aspect of all these trends is the enforcement of reusability and shareability of products or components based on new technology standards. DMSO HLA makes the first major step in this direction by offering the interoperability framework between a broad spectrum of simulation paradigms, including both real-time and logical time models (DMSO 1998).

However, HLA standard specification leaves several implementation decisions open and to be made by the application developers - this enables reusability and integrability of existing codes but often leaves developers of new simulations without enough guidance. In WebHLA, we fill this gap by using the emergent standards of Web based distributed computing - we call it Pragmatic *Object Web* (Orfali 1998; Fox et al. 1999) - that integrate Java, CORBA, COM and W3C WOM models for distributed componentware.

Traditional HPCC, dominated by data parallel MPP didn?t make significant inroads into the DoD M&S where the focus is on task parallel heterogeneous distributed computing. Recent trends towards commodity based HPCC systems such as NT clusters offer a new promising framework for new generation high performance high fidelity M&S environments such as addressed by JSIMS, JWARS, JMASS or Wargame2000 programs.

We therefore believe that WebHLA, defined as the convergence point of the standardization processes outlined above will offer a powerful modeling and simulation framework, capable to address the new challenges of DoD computing in the areas of Simulation Based Design, Testing, Evaluation and Acquisition.

The overall architecture of our WebHLA prototype follows the 3-tier architecture of our *Pragmatic Object Web* (Fox et al. 1999) (see Figure 1) with the RTI-over-JWORB based middleware, backend simulation modules, wrapped via CORBA/COM as FOM or SOM objects and with the WebFlow/DirectX based visual authoring / runtime front-ends.

In the following, we describe in more detail in Chapter 2 the WebHLA components listed above, followed by a brief overview in Chapter 3 of the emergent WebHLA application domains such as: Distance Training, Resource Management for Metacomputing and/or Commodity Clusters, and Simulation Based Acquisition.

2. WebHLA COMPONENTS

2.1 Object Web RTI

Current HLA is a custom distributed object model but DMSO?s longer range plan includes transferring HLA to industry as CORBA Facility for Modeling and Simulation.

Anticipating these developments, we have recently developed in one of our HPCMP FMS PET projects at NPAC an Object Web based RTI (Fox et al.1998c) prototype, which builds on top of our new JWORB (Java Web Object Request Broker) middleware integration technology.

JWORB is a multi-protocol Java network server, currently integrating HTTP (Web) and IIOP (CORBA) and hence acting both as a Web server and a CORBA broker. Such server architecture enforces software economy and allows us to efficiently prototype new interactive Web standards such as XML, DOM or RDF in terms of an elegant programming model of Java, while being able to wrap and integrate multi-language legacy software within the solid software engineering framework of CORBA.

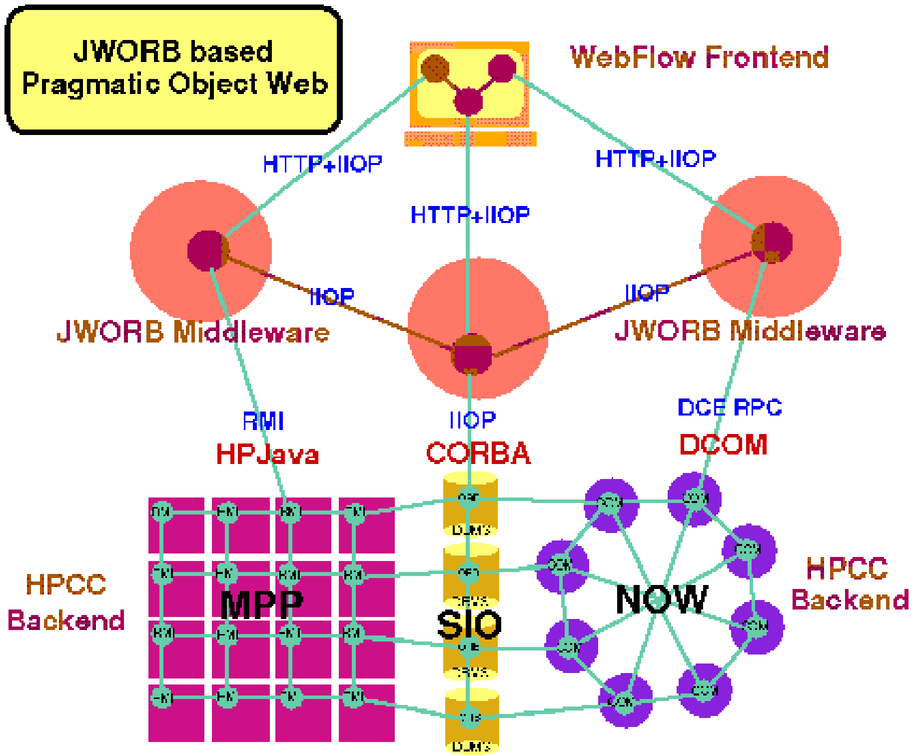

Figure 1 *Illustration of the communication protocol integration within our JWORB based Pragmatic Object Web. JWORB uses Java to integrate HTTP with IIOP and then it connects with NT clusters via COM/CORBA bridge.*

We are now testing this concept and extending JWORB functionality by building Java CORBA based RTI implementation structured as a JWORB service and referred to as *Object Web RTI* (Figure 2). Our implementation includes two base user-level distributed objects: RTI Ambassador and Federate Ambassador, built on top of a set of system-level objects such as RTI Kernel, Federation Execution or Event Queues (including both time-stamp- and receive-order models). RTI Ambassador is further decomposed into a set of management objects, maintained by the Federation Execution object, and including: Object Management, Declaration Management, Ownership Management, Time Management and Data Distribution Management (see Figure 9).

To be able to run C++ RTI demo examples, we developed a C++ library which: a) provides HLA RTI v1.3 C++ programming interface; and b) it is packaged as a CORBA C++ service and, as such, it can easily cross the language boundaries to access Java CORBA objects that comprise our Java RTI. Our C++ DMSO/CORBA glue library uses public domain OmniORB2.5 as a C++ Object Request Broker to connect RTI Kernel object running in Java based ORB and runs on Windows NT and Sun Solaris 2.5 environments. Through library, user defines the RTIambassador object as usual but in the implementation it actually accesses the OWRTI CORBA service and executes each call on this service. Similarly, user supplied federate ambassador object is managed by a CORBA Object which resides at the client side and forwards all the call it received from RTI to the user's federate ambassador object.

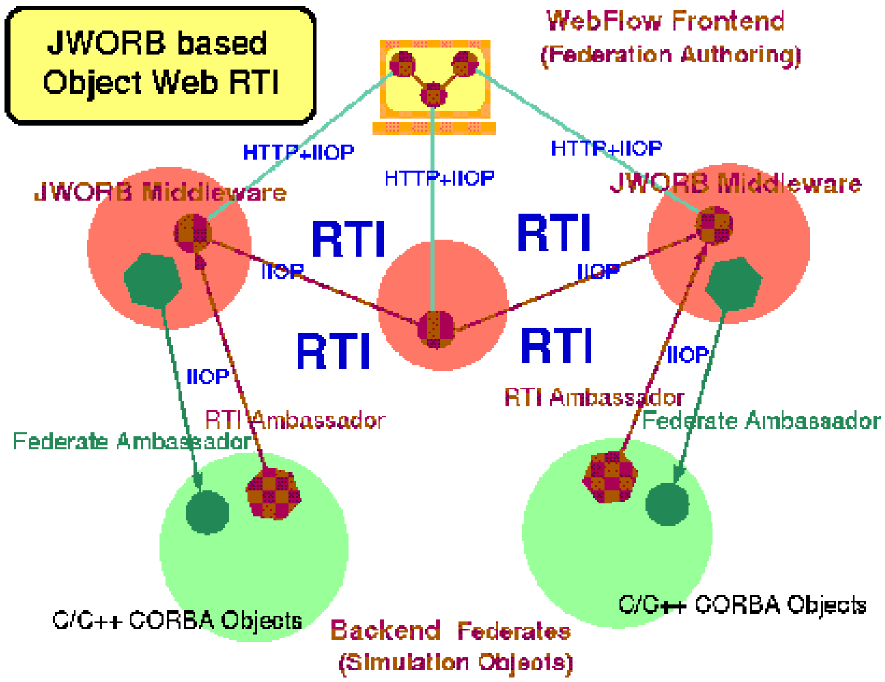

Figure 2 *Top view representation of Object Web RTI: RTI Ambassador is Java CORBA object maintained by JWORB middleware; Federate Ambassador is (typically C++) CORBA object maintained by the backend federate; WebFlow front-ends tools are available for visual authoring of federation configuration.*

2.2 Visual Authoring Tools for HLA Simulations

DMSO has emphasized the need to develop automated tools with open architectures for creating, executing and maintaining HLA simulations and federations. The associated Federation Development Process (FEDEP) guidelines enforce interoperability in the tool space by standardizing a set of Data Interchange Formats (DIF) that are mandatory as input or output streams for the individual HLA tools. In consequence, one can envision a high-level user friendly e.g. visual dataflow authoring environment in which specialized tools can be easily assembled interactively in terms of computational graphs with atomic tool components as graph nodes and DIF-compliant communication channels as graph links (Figure 3).

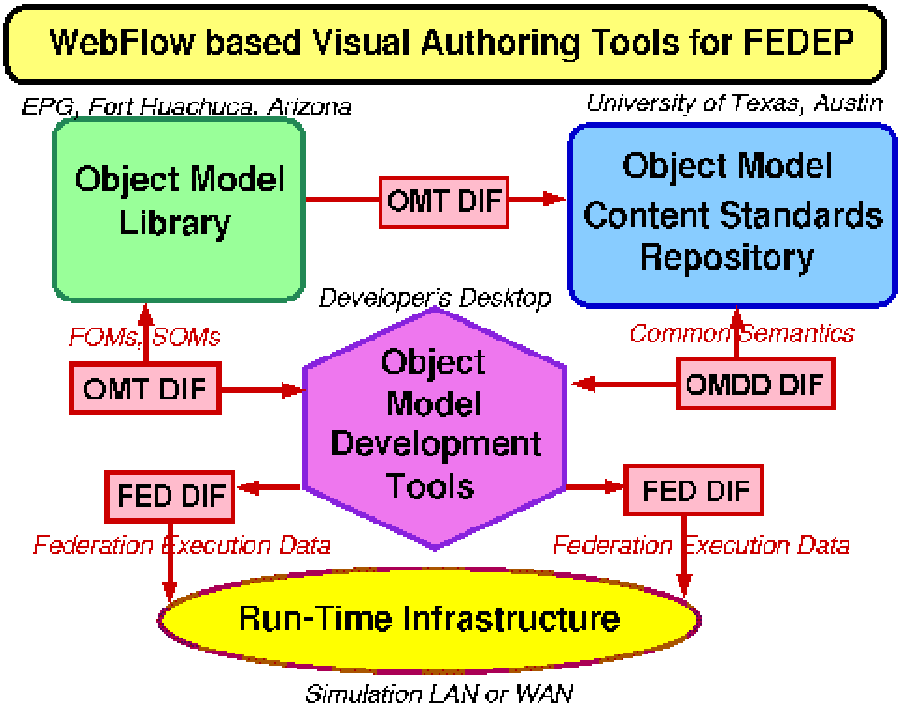

Figure 3 *WebFlow based representation of DMSO FEDEP: individual FEDEP tools are mapped on WebFlow modules and the DIF files are mapped on WebFlow communication channels.*

To provide a natural linkage between DMSO DIF and the emergent Web data interchange format standards to be formulated within the XML framework, we are constructing a trial mapping between FED and XML, which is used in our current generation tools. In particular, we defined a trial version of HLAML for Federation configuration file definition. We developed two converter programs, which read HLAML and produce DIF or HTML file. HLAML is defined via XML DTD so that we were able to use free XML parsers. Current OWRTI uses this file format for federation definition files. Our OMBuilder tool discussed in the following also supports the same file formats so that we can develop federations using this tool and import them easily into our HLA RTI framework.

Within our HPCMP FMS PET project at NPAC we are building visual HLA tool assembly framework (Fox et al. 1998a) illustrated in Figure 3 on top of the NPAC WebFlow system (Bhatia et al. 1997).

WebFlow is a Web/Java based visual dataflow environment with the Web browser based computational graph editor and the runtime given by a mesh of interconnected Java Web Servers, used to manage WebFlow middleware module wrappers and acting as proxies to the backend computational modules (Figure 4).

We started our visual WebHLA authoring project by analyzing currently existing tools in this area. In particular, we inspected the Object Model Development Tool (OMDT) by Aegis Research Center, Huntsville, AL as a representative current generation DMSO FEDEP tool. OMDT is a Windows 95/NT-based application that supports building HLA Object Model Template (OMT) tables such as Class, Interaction or Attribute Tables using a spreadsheet-like user interface. We found OMDT useful in the standalone mode but not yet ready to act as a standardized reusable component in larger toolkits. We therefore decided to build an OMDT-like editing tool based on Microsoft Component Object Model (COM) architecture.

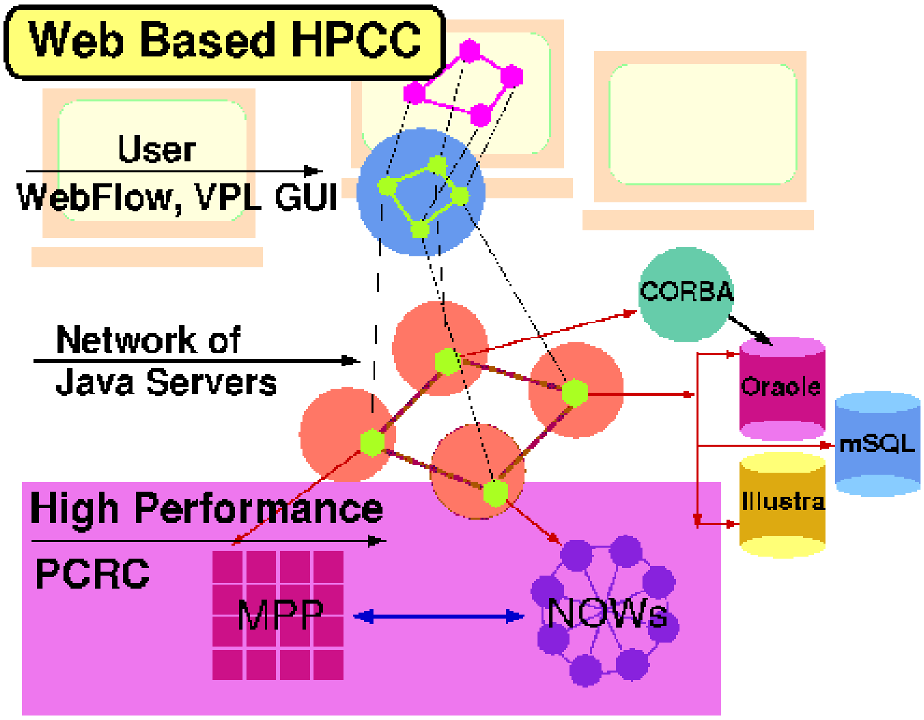

Figure 4 *Top view of the WebFlow system: its 3-tier design includes Java applet based visual graph editors in tier 1, a mesh of Java servers in tier 2 and a set of computational (HPC, Database) modules in tier 3.*

Rather than building our sample tool from scratch, we construct it by customizing Microsoft Excel Component using the Visual Basic for Applications and the OLE automation methodology. Using this approach, we were able to emulate the look-and-feel of the OMDT tool (see Figure 5), while at the same time packaging our tool, called OMBuilder, as a reusable COM or ActiveX component that can smoothly cooperate with other visual authoring tools within the WebFlow model (Figure 6). We also extended the OMDT functionality in OMBuilder by adding support for initializing and runtime steering the simulation attributes and parameters.

Next, we constructed a proof-of-the-concept demo that offers WebFlow and OMBuilder based visual authoring toolkit for the Jager game (available as part of the DMSO RTI distribution). Each Jager player (both human and robot) is represented by a suitable visual icon in the WebFlow editor.

A particular game scenario is constructed by selecting the required player set and registering / connecting them as nodes of a suitable connected graph. We also constructed a 3D viewer, operating in parallel with the standard 2D Jager viewer and used for experiments with parameter initialization via the extended OMBuilder editor (Figure 6).

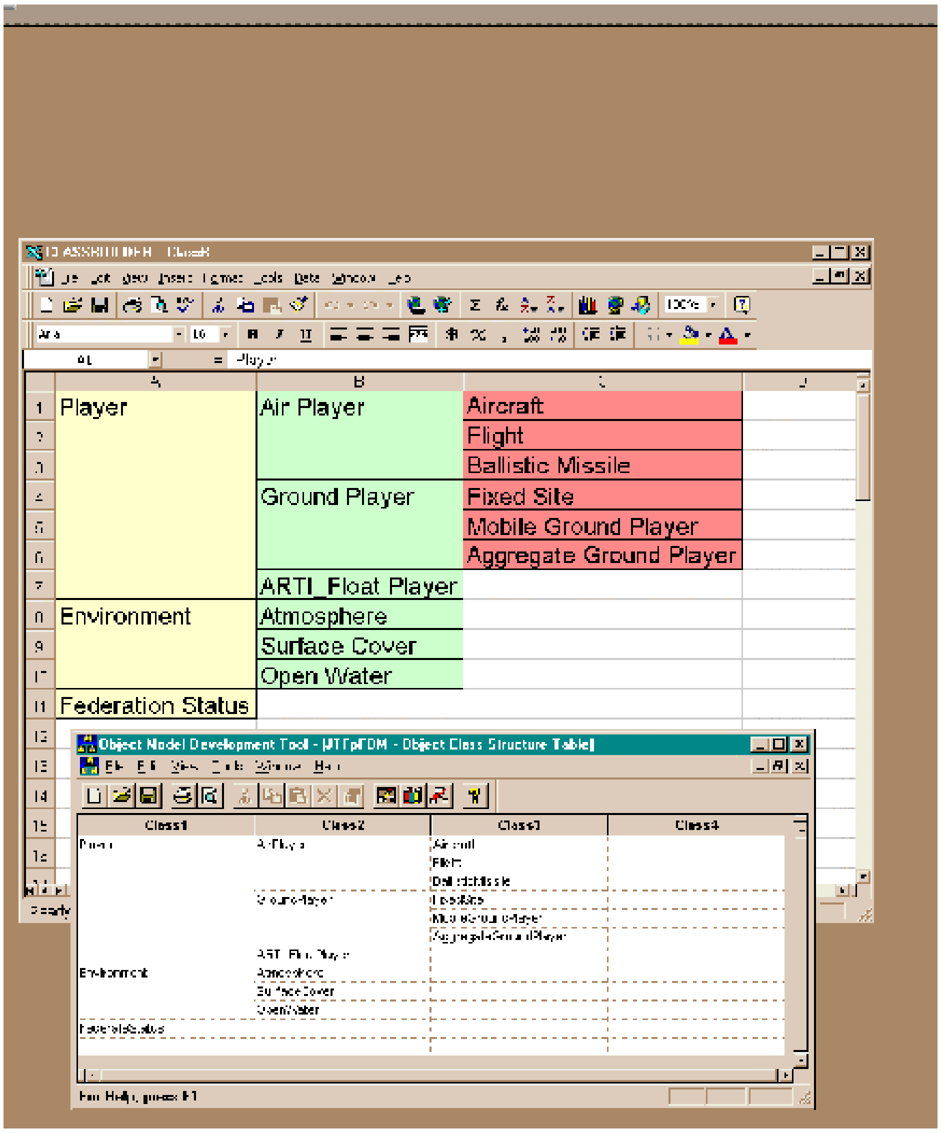

Figure 5 *OMDT by Aegis Corporation (front window) compared with our OMBuilder (back window): both tools have similar functionality and look-and-feel but OMBuilder is constructed via VBA scripting on top of Microsoft Excel.*

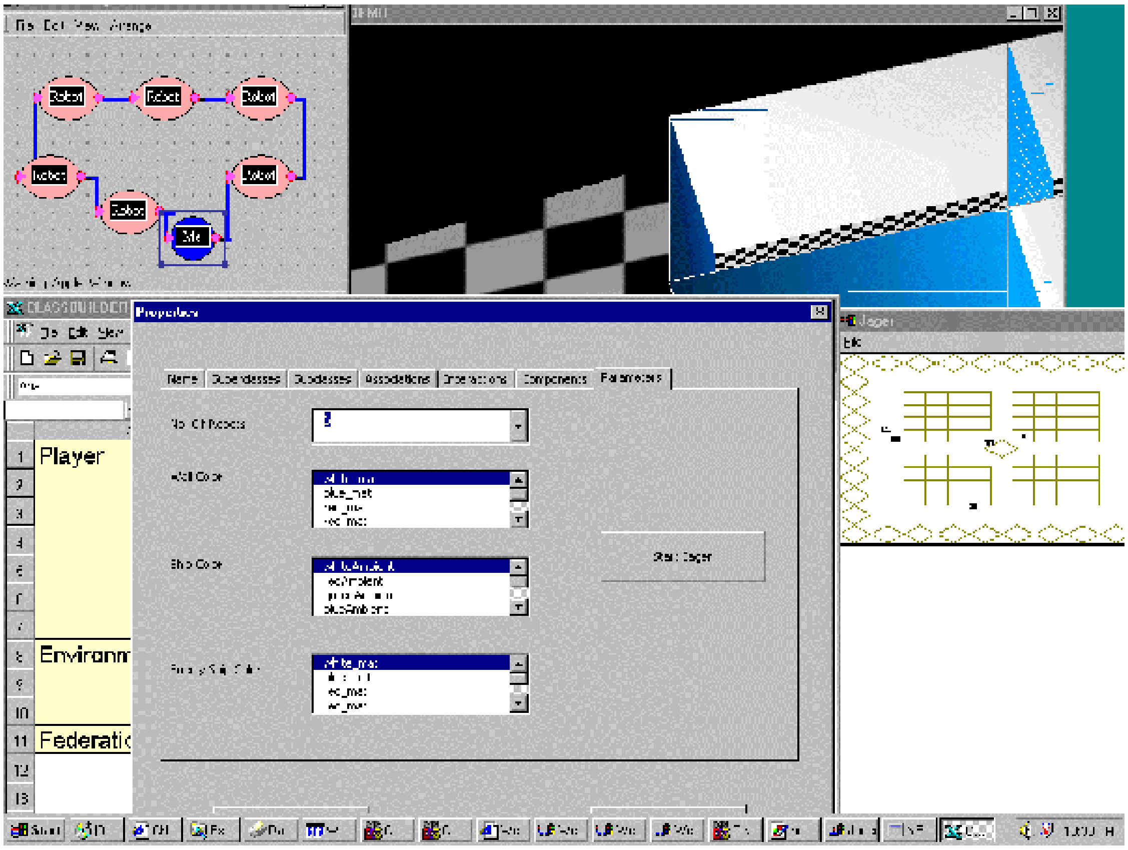

Figure 6 *Proof-of-the-concept visual authoring tools for Jager. WebFlow editor (upper left) allows specifying the configuration for the Jager game players. Both the original 2D (middle right) and our trial 3D viewer (upper right) are displayed. A runtime extension of the OMBuilder tool (lower left) allows the user to initialize, monitor and steer the simulation or rendering parameter (lower middle).*

2.3 Parallel ports of selected M&S modules

In parallel with prototyping core WebHLA technologies described above, we are also analyzing some selected advanced M&S modules such as the CMS (Comprehensive Mine Simulator) system developed by Steve Bishop?s team at Ft. Belvoir, VA that simulates mines, mine fields, minefield components, standalone detection systems and countermine systems including ASTAMIDS, SMB and MMCM. The system can be viewed as a virtual T&E tool to facilitate R&D in the area of new countermine systems and detection technologies of relevance both for the Army and the Navy. We are currently constructing the parallel port of the system to Origin2000. In the next step, we intend to use such Parallel CMS as one of the WebHLA backend modules for Metacomputing FMS simulations, discussed in Chapter 3.

2.4 Database and data mining back-ends

Our database activities for the WebHLA backend include both research into the emergent transparent persistency models such as CORBA PSS, OLEDB, JDBC or XML/RDF (Fox et al. 1998b), and the practical development of specific relational databases tailored for the individual components, projects and customers. For example, we are building SQL databases with the software documentation and PowerPoint presentation for the FMS Training Space and the distributed resources databases for the Metacomputing FMS and Commodity Clusters. We are also interacting with the Virtual Proving Ground

(VPG) team at ARL/ATC in Aberdeen, MD where we currently start providing help with the data mining services for the T&E historical test data.

2.5 Realtime multiplayer gaming front-ends

In our Pragmatic Object Web approach, we integrate CORBA, Java, COM and WOM based distributed object technologies. We view CORBA and Java as most adequate for the middleware and backend, whereas COM as the leading candidate for interactive front-ends due to the Microsoft dominance on the desktop market.

Of particular interest for the M&S community seems to be the COM package called DirectX which offers multimedia API for developing powerful graphics, sound and network play applications, based on a consistent interface to devices across different hardware platforms. DirectX comes as a set of APIs, each of which cater to a different aspects of an application being developed - DirectDraw for graphics, DirectSound for adding sound, DirectPlay for networked gaming etc. Of these, the DirectPlay API is of interest to us while discussing HLA/RTI, because of it's similarities with the RTI.

Due to the design proximity, we could naturally create an HLA simulation using DirectX, while still using RTI as a communication medium. The simulation would thus conform to the HLA/RTI standards and perform well in the PC environment. Our proof-of-the-concept demonstration of such a linkage between Microsoft DirectX and Object Web RTI on a real-time basis is using a modified version of the popular DirectX game - *Donuts* as the front end. The objective here, apart from presenting OWRTI, was to demonstrate the ability to bring together the high-end RTI and commodity technologies like DirectX to build HLA compliant simulations. The whole set-up of the demo included a wide range of technologies like CORBA, Java, C++ and DirectX/COM.

The entire front end of the demo was built using Microsoft's DirectX API for gaming & multimedia. The basic structure of the sample game *Donuts* (provided with the Microsoft DirectX SDK) was used and suitably modified to incorporate an RTI layer for communication between two remote games. The game involves shooting down *donuts* using a gun/ship, which can be manipulated around the screen by the user.

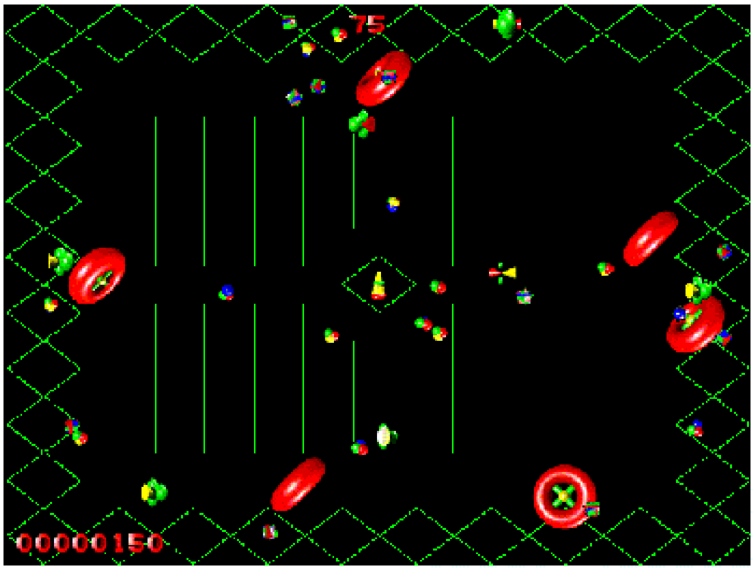

Figure 7 *Sample screen from multi-player Jager Donuts demo using modified Donuts front-end (to conform to the Jager game geometry) and the Object Web RTI based multi-user communication.*

The *remote donuts* game, as a first and simple step, sends and responds only to RTI interactions. A game (federate) responds to an interaction from another federate by putting a donut on the screen. A federate can send an interaction to other federates on the command of the user. The RTI server forwards the interaction to all other federates in the federation, each of which would respond by putting a donut on their screens.

In the demo, we provided an automatic interaction generator. This is another federate joining to the Federation and producing only interactions. Its parameters are controlled through a Web interface. A servlet is responsible for getting number of interactions to be produced, the time interval between interactions and message order type (receive/timestamp). Then it forks a separate federate which is written in Java and interacts with the RTI. As a result, user automatically starts receiving messages from this federate.

With the DirectX platform and the associated high level Web tools such as Chromeffects, Microsoft enters the computer gaming market. Hence, once can expect the coming explosion of Web based networked games for Windows. Figure 8 illustrates a state-of-the-art DirectX based real-time commodity graphics scene in a professional game Art of Flying by Blue Moon Interactive. Our DirectX-OWRTI bridge discussed above will allow us to reuse these entertainment front-ends for the military training purposes by coupling them with the HLA compliant simulation backends and delivering the training in the Web based interactive mode.

3. Emergent WebHLA Applications

In parallel with developing focused WebHLA components discussed in Chapter 2, we also start identifying some promising WebHLA application domains and we initiate the design and prototyping activities. In this Chapter, we present a brief overview of our current activities in the following emergent WebHLA application areas: Distance Training, Resource Management for Metacomputing and/or Commodity Clusters, and Simulation Based Acquisition.

Figure 8 *Sample screen from the DirectX based professional game Art of Flying Distance Training We are building WebHLA based interactive training on the use of advanced M&S kernels such as SPEEDES. Students, teachers are the target systems in this approach are represented as federates and collaborate within a training FOM. Web and commodity gaming offer natural front-ends for such online multiplayer training environments.*

Metacomputing FMS We are working with the Night Vision Lab at Ft. Belvoir, VA on building high performance version of their CMS (Comprehensive Mine Simulation) system to enable support for large minefields of order of 1,000,000 and more mines. Such large scale battlefields admit natural metacomputing implementation with geographically distributed and WebHLA coordinated federates representing minefields, their components, propagating vehicles etc. One such early demonstration with simulation nodes in Maryland, New York and Florida is being prepared for Supercomputing 98, Nov 98.

Cluster Management RTI can be viewed as a high level abstraction of a distributed operating system with machines / nodes represented as federates, clusters as federations, with time management responsible for job scheduling, ownership management linked with security and so on (see Figure 9). We are currently exploring Object Web RTI as such a high level operating and control framework for managing heterogeneous and growable commodity clusters including NT, UNIX and LINUX nodes.

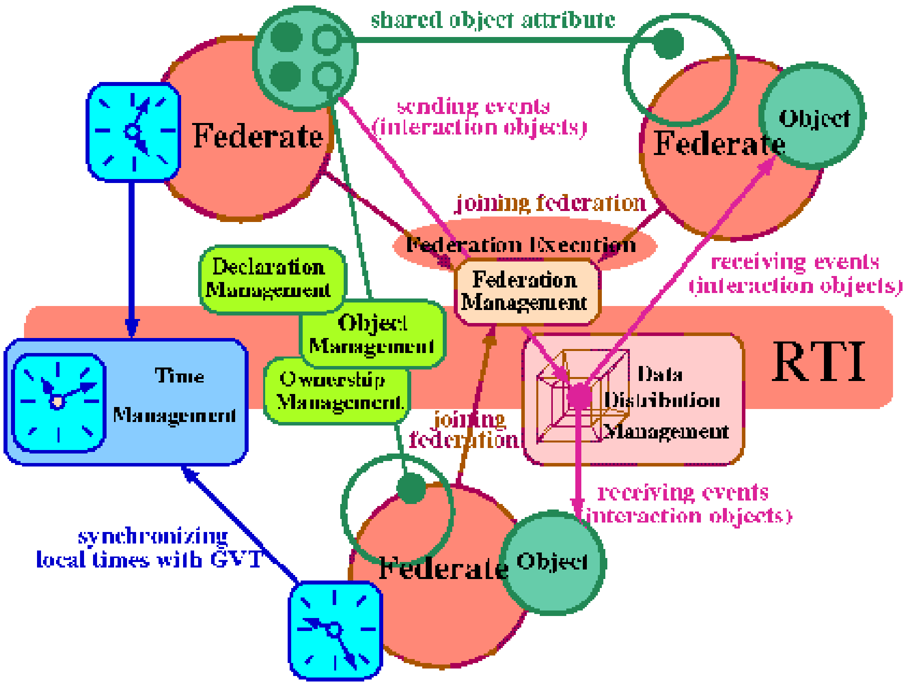

Figure 9 *Distributed object based architecture of DMSO RTI - originally constructed for the M&S domain but naturally extensible for other distributed computing management services such as cluster, metacomputing, collaboration or distance training management discussed in the text.*

Simulation Based Acquisition Figure 10 illustrates our envisioned end product in the WebHLA realm - a distributed, Web / Commodity based, high performance and HLA compliant Virtual Prototyping Environment for Simulation Based Acquisition with Object Web RTI based software bus, integrating a spectrum of M&S tools and modules, wrapped as commodity (CORBA or COM) components and accessible via interactive Web browser front-ends. Such environments, currently operated only by large industry such as Boeing or Automotive, become affordable within the current technology convergence process as envisioned in first section and quantified in our WebHLA integration program.

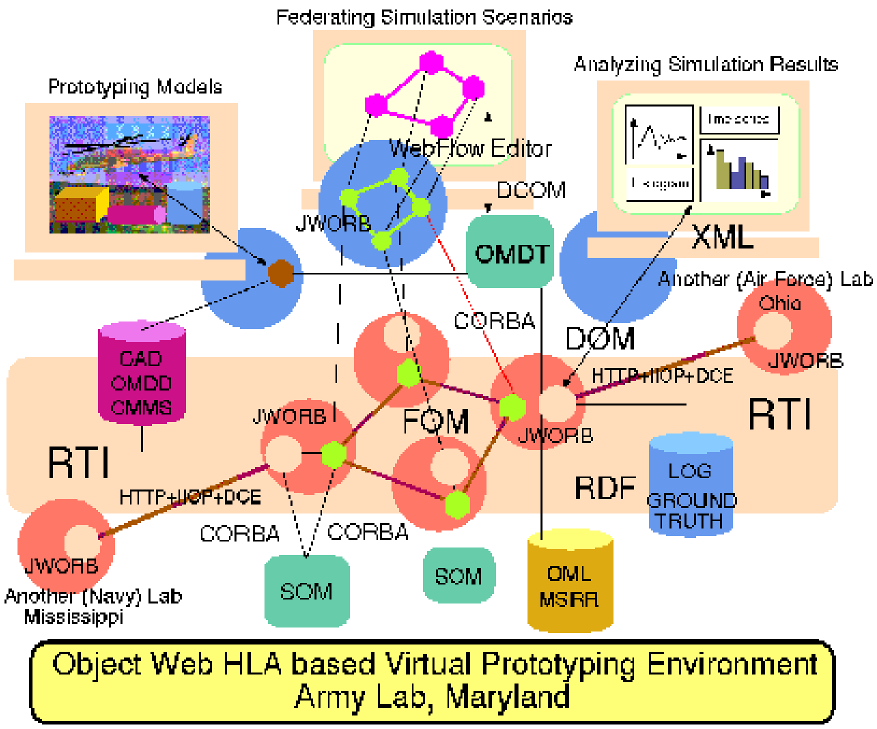

Figure 10 *Overall architecture of a planned WebHLA based Virtual Prototyping Environment for Simulation Based Acquisition: Object Web RTI acts as a universal software bus, managing a spectrum of resources for simulation, clustering, collaboration, training, data mining, data warehousing etc.*

Indeed, the challenge of SBA is to successfully integrate M&S, T&E, HPC, Metacomputing, Commodity software for Real-Time Multimedia front-ends and Database back-ends, Collaboration, Resource Management and so on - these capabilities represent individual WebHLA components discussed in the paper and being now prototyped in a coordinated, Pragmatic Object Web based development and integration framework.

4. SUMMARY

We presented here our approach towards building WebHLA as an interoperable simulation environment that implements new DoD standards on top of new Web / Commodity standards. Our initial results are encouraging and we believe that WebHLA will evolve towards a powerful modeling and simulation framework, capable to address new challenges of DoD and commodity computing in many areas that require federation of multiple resources and collaborative Web based access such as Simulation Based Design and Acquisition, Distance Training or Online Multiplayer Gaming.

5. REFERENCES

Defense Modeling and Simulation Office (DMSO) 1998, High Level Architecture, http://hla.dmso.mil.

D. Bhatia, V. Burzevski, M. Camuseva, G. Fox, W. Furmanski and G. Premchandran, 1997, WebFlow - a visual programming paradigm for Web/Java based coarse grain distributed computing, Concurrency: Practice and Experience, vol. 9, no. 6(Jun.), 555-577.

Robert Orfali and Dan Harkey, 1998, Client/Server Programming with Java and CORBA, 2nd Edition, Wiley.

G. C. Fox, W. Furmanski, H. T. Ozdemir and S. Pallickara, 1999, Building Distributed Systems for the Pragmatic Object Web, Wiley, (in progress).

G. C. Fox, W. Furmanski, B. Goveas, B. Natarajan and S. Shanbhag, 1998, WebFlow based Visual Authoring Tools for HLA Applications , In Proceedings of ITEA?98 (Aberdeen, MD, July 98).

G. C. Fox, W. Furmanski and T. Pulikal, 1998, Evaluating New Transparent Persistence Commodity Models: JDBC, CORBA PPS and OLEDB for HPC T&E Databases , In Proceedings of ITEA?98 (Aberdeen, MD, July 98).

G. C. Fox, W. Furmanski and H. T. Ozdemir, 1998, Java/CORBA based Real-Time Infrastructure to Integrate Event-Driven Simulations, Collaboration and Distributed Object / Componentware Computing, In Proceedings of PDPTA?98 (Las Vegas, NV, July 98).