Web-Based Operation of Interactive Sharable Environment for Collaborative Spacecraft Design

Norman Lamarra and Julia Dunphy
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr, M/S 126-201
Pasadena, CA 91109
Norm.Lamarra@jpl.nasa.gov (818)393-1561
Julia.Dunphy@jpl.nasa.gov (818)393-3565

Abstract— A web-based interface has recently been added to an advanced integrated environment being developed at JPL that links collaborators who wish to perform interactive design analyses and/or mission simulations. The environment, described previously, utilizes commercial technology (such as 3D visualization) where applicable, but key pieces are currently provided by software developed in-house, e.g., for user interaction and spacecraft modeling. Within the environment, a mission scenario can be built and exercised at various levels (e.g., macro or micro simulation, modeling or analysis), and existing tools preferred by participants can be integrated "in-place". Mission information (e.g., target body, space environment), spacecraft information (e.g., drawings, structures), and payload information (e.g., subsystem or instrument models) can be connected into a simulation which can now be executed by collaborators using only their web browsers. In addition, web-based tools provided by collaborators can be integrated into the environment and remotely executed. This allows interaction of the users with components of particular interest to each, while others can view the "big picture" results of the interactions, and make recommendations such as parameter trades or component alternatives. Components of this environment are currently being developed by several NASA centers who wish to leverage each other's strengths; and a shared information infrastructure facilitates the connections (e.g., access to databases of designs, products, models and data). We believe that the collaborative process is most successful when the participants can immediately see the collective results of their separate inputs; therefore our goal was to facilitate real-time collaborative interactivity. We discuss the problems and achievements from early utilization of this evolving interactive environment.

TABLE OF CONTENTS

1. INTRODUCTION

Background

At last year's AESS meeting, we described the concept and implementation for an Interactive Sharable Design Environment (ISDE) [1]. This environment consists of a central server, several agents, and interfaces for project editing and simulation execution. The central server (the Millennia Engine) was described in some detail in that paper, along with the X-based editing interface (Methedit), and the OpenGL-based execution interface (built using a Virtual-Reality development environment from MuSE Technologies in Albuquerque, NM; www.musetech.com). Both interfaces were built as CORBA clients of the Millennia Engine (CORBA server). This paper describes the efforts over the past year to add two additional browser-based user interfaces to the server, thus providing low-cost (free) collaborative access to the design environment from almost any platform. Before describing these enhancements, we shall briefly review the ISDE's concept and features.

Problem description

In the NASA/JPL context (robotic exploration of deep space), the trend is towards planning smaller missions with much more limited goals, and requiring less long-term budgetary commitment. By funding many such smaller missions, like Mars Pathfinder (e.g., $100-200M each), rather than large-scale missions like Cassini (e.g., $1-$2B each), NASA expects an improved aggregate return on investment, and a reduction in the consequences of catastrophic failure, such as loss of a spacecraft. Unfortunately, less time, funding, and workpower is also available to implement each such small-scale mission; therefore, it becomes increasingly important to reduce estimated mission lifecycle cost and risk, both before mission selection and during development.

Goals

Given the requirements and problems described in the previous paper [1], the specific goals of the work on the ISDE are repeated here for convenience:

a) to develop an integrating framework to facilitate collaboration between members of a geographically-distributed design team;
b) to demonstrate use of this integrating framework in real space mission design scenarios;
c) to explore innovative ways for naïve users to interact with the environment to achieve their design tasks (in particular, voice interaction, virtual reality, and intelligent agent assistance);
d) to provide ways of integrating existing and new Design Resource Products (DRPs), such as design tools or data, into the environment;
e) to leverage concurrent work in distributed-system technologies (such as distributed simulation) performed at JPL and elsewhere.

More specifically, we identified and attempted to address the following issues:

a) ability to assemble a dynamic mission simulation from models, which may initially be imprecise and incomplete;
b) ability to integrate with mission planning tools sufficiently to provide operational context (such as trajectory, sequencing) for critical mission phases (such as entry-descent-landing or orbital science observation);
c) integration of analysis tools to evaluate the operational behavior of the system during such phases;
d) visualization of key measurables, such as science observations, system performance, or physical layout;
e) ability to allow interactive modification of scenario, system, or subsystem components (such as position or performance parameters) in order to facilitate design trades;
f) ability to reduce cost by assembling a "virtual team" of diverse experts without necessitating collocation.

2. Approach

The proposed approach was to leverage work being performed at JPL in distributed interactive simulation, visual programming, and intelligent tools. This work began several years ago with the Multidiscplinary Integrated Design Assistant for Spacecraft (MIDAS), then became the ISDE. Most recently, the work performed over the last year was focused primarily on web collaboration via a browser client. Section 3 describes the design and architecture of the prototype ISDE, including the migration of its capabilities to the web to facilitate collaboration and universal accessibility. Section 4.0 presents some Example User Scenarios.

Overview/Review

MIDAS was developed at JPL to integrate distributed tools and present the user with a "plug and play" interface that allows many of the goals of *Integrated Design Systems* to be met. MIDAS provides access to a distributed database of components, analysis tools, visualization tools, drawings, and documents. A designer is able to access these items and seamlessly use them to come up with an optimized design methodology. The methodology is generated graphically (producing a "methogram"), and is intended to describe the process that the designer is accustomed to following when making decisions about the form and attributes of each component. The methogram can be saved in the database and reused either in another part of the design or in a subsequent similar design. Capture and reuse of the design process in this way is akin to Knowledge-Based Engineering (KBE).

Straightforward processes that must be repeated each time a new design comes along are ideal candidates to be transferred into methograms. For example, a certain structural shape may be designed by choosing beginning dimensions, running a thermal analysis, choosing a new dimension based on the results of the analysis, and iterating the process until satisfactory analysis results are achieved. These steps may be captured in a methogram that can then be executed on a supercomputer or a network of workstations automatically.

After the methogram is designed, it needs to be debugged (using MIDAS facilities) until the operator feels that the methogram is as general as possible, yet contains sufficient detail to be useful.

A designer, who may or may not be the originator of the methogram, can then provide the input, in the form of design requirements or performance specifications, and execute the methodology to arrive at the point design. If the requirements change (as they nearly always do) the designer can re-execute the parametrized process in a matter of minutes instead of having to go through all the design steps laboriously.

3. The Interactive Sharable Design Environment

The central component of the ISDE is a Programmable Tool Server (the Millennia Engine), which enables these models and tools to be interconnected. This allows distributed real-time simulation to be performed at various levels of fidelity under user control. The environment allows commercial tools to be used as well as experimental or "home-grown" tools (such as probabilistic analysis methods). The components of the ISDE are pictured schematically in Figure 1, which also shows current participation of various NASA centers and vendors.

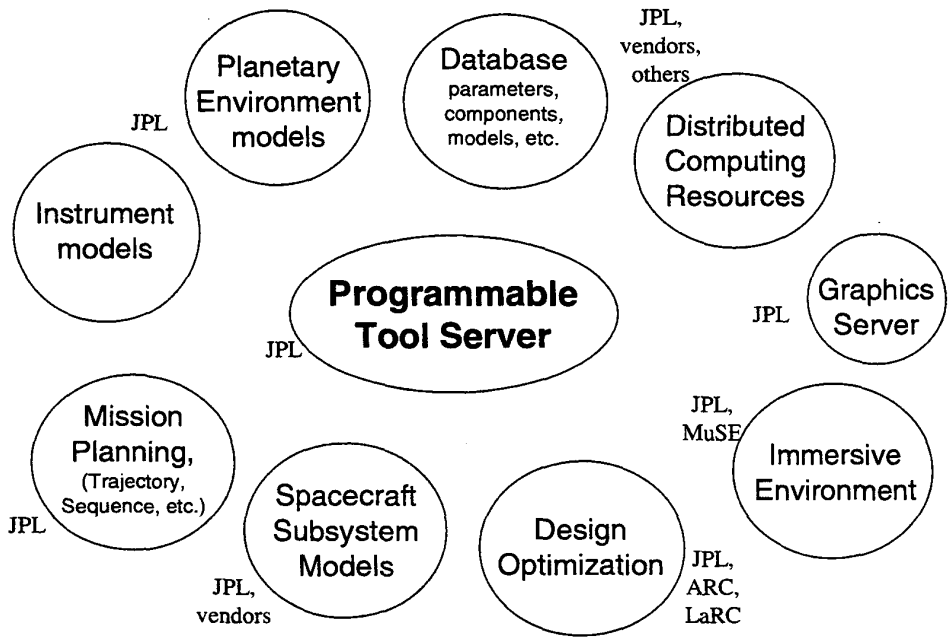


Figure 1: Components of Interactive Sharable Design Environment

Millennia Engine and the Web

While strictly not related to Web Based Collaboration, a related topic is the use of web ready design tools in a Millennia Engine project. Normally the engine starts up and passes data to tools on other platforms via remote procedure calls (RPCs) or by the use of a slightly higher-level cross-platform interconnection methodology called Parallel Virtual Machine (PVM). These protocols are usually only available on "friendly" computers, normally those co-located with the Millennia Engine platform.

These days, because of the enormous web activity, many tools are becoming available at "unfriendly" sites that do not allow direct access (e.g., via RPCs or PVM) because of security concerns. These sites typically allow their tools to be used in a structured way, e.g., by setting up web forms for data input, and displaying the tool results by dynamically creating a new web page, VRML file, or similar viewable products. We recognized that the Millennia Engine could be much enhanced if it were able to communicate with such sites. The possibilities opened up by this enhancement include being able to pass data generated by web-based tools to those on local machines and back again. We have actually constructed a methogram for NASA's Ames Research Center (ARC) with over a dozen distributed web tools all chained together in this way.

Addition of this "web tool" feature to the Millennia Engine (Version 6.0) was quite difficult. Coincidentally, the World Wide Web Consortium (W3C) was just releasing a C++ library package to allow communication with a web site from a C++ program emulating a user interaction. However, our first few attempts at using this library showed up several important bugs that the W3C acknowledged and took care of very efficiently. Eventually we achieved success. We decided to make the web communication process a separate CORBA server (web_agent) so that other projects could use it without having to integrate it directly into their code.

NASA-ARC provided us with a challenge to try using several web sites at LaRC, JPL, and ARC for the collaborative design of aeroshells. This involves aerodyamic (LaRC) and thermal protection (ARC) for situations such as robot landers on Mars. The methogram we constructed will be made available to all NASA sites through netMethedit (see Figure 2) and perhaps netISDE (see below). Through the use of netMethedit, it will be possible to edit the methogram to substitute other web site tools or tools on local machines.

Browser-Based Methogram Editing: netMethedit

The methogram editor tool, netMethedit, is a rewrite of Methedit in Java. Since Methedit communicates with the Millennia Engine using CORBA, it was also necessary to add Java classes that can perform the same function. Fortunately, a new Java/CORBA package became available in 1996 called OrbixWeb (IONA Technologies). A great effort was made to keep the look and feel of Methedit in the new product. The maturity of the Java AWT package presently offers all the features available in the older X-windows package used by the original Methedit and we were able to keep most of the interface identical. One area where we had problems was in terms of design security. In a non-browser client (e.g., via xterm), the Millennia Engine can determine who is connected by asking the operating system to give information about the login. This is not possible with "anonymous" browser access so we plan to add extra username/password protection in a future release.

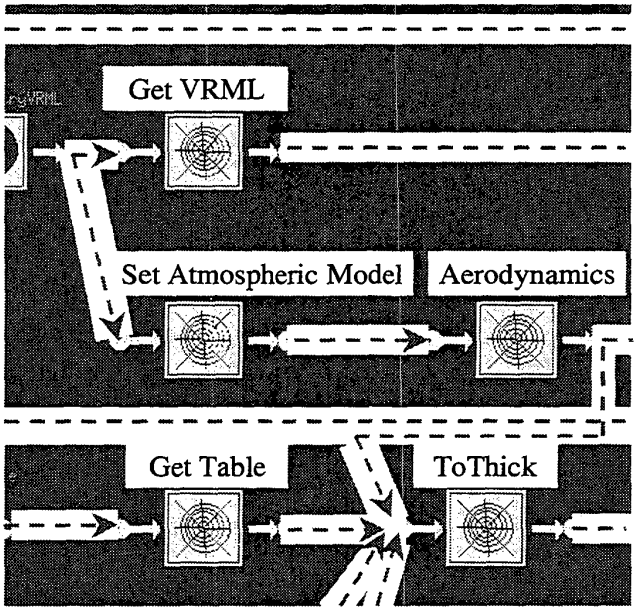


Figure 2: Portion of methogram showing data flow between web-based tools

Browser-based 3D interface: netISDE

The net version of the ISDE, netISDE, presented a different challenge. The MuSE software library can not be used in the same way as it is when running OpenGL directly on an SGI or Sun Microsystems workstation. However, the rest of the ISDE software is capable of being run from a browser. It was therefore decided to re-organize the ISDE software as shown in Figure 3, so that it can operate either with MuSE providing display and navigation functions, or with separate code modules employing Virtual Reality Markup Language (VRML) for the display and Java for the navigation. This re-organization has provided a significant side benefit in improved maintainability, since netISDE and ISDE share over 80% of their C++ code and only differ in graphics functions.

The programming task condensed to writing a scene navigator package in Java to display on the browser on top of the scene that is rendered in VRML. Since some navigation functions, such as adjusting viewing angles and viewport sizes, are really local actions, it was decided to add these features by using direct interaction with the browser in the form of commands to the VRML rendering package (COSMO Player from SGI).

The data flow between a user action and the result is very complex in this design, as shown in Figure 4. When a user presses a button in the Java navigator, the Java program makes a URL "POST" call to a Practical Extraction And Report Language (PERL) script on the machine which runs the Millennia Engine. This PERL script then starts up the ISDE version containing the VRML scene generator package. The ISDE program communicates with the Millennia Engine using CORBA calls as usual to perform the user request. When the user action is complete, the ISDE generates a new scene in VRML and exits. The calling navigator program then does a URL "GET" call to the machine running the Millennia Engine and displays the VRML, as shown in Figure 4. While not all the MuSE navigation features are yet available, most of the most frequently used features are now operational.

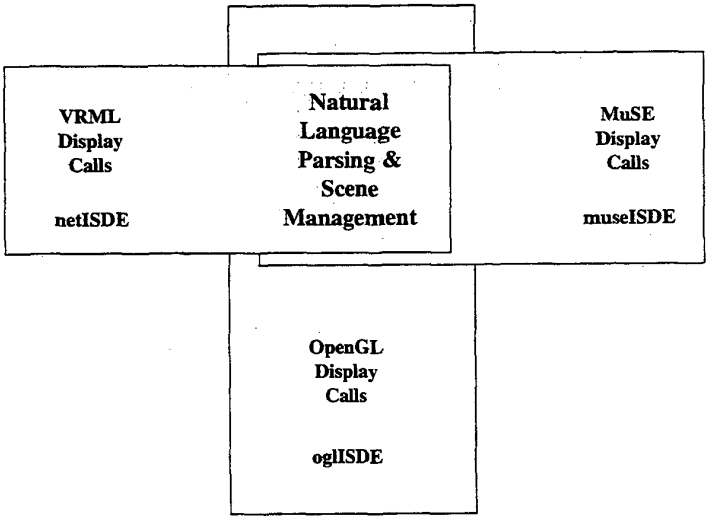


Figure 3: ISDE interface configurations

4. Example User Scenarios

We released the first version of netMethedit in September 1998 and NASA-ARC is our first user. The JPL/ARC/LaRC aeroshell design collaboration is shown in Figure 5 below, and connects approximately 15 tools together in a web-based methogram, a portion of which was shown in Figure 2. This includes local FORTRAN code running at JPL on a PC, executed via a freely-available web server (Apache) on that PC. Previously, we described future possible single-user and collaborative design scenarios using an immersive ISDE. With the web versions of netMethedit and netISDE, some of these desired modes of collaboration are now possible. For example, when a single user connects a browser to the netMethedit URL, he is connected to a "session broker" that asks for username/password and operating mode desired (standalone or collaborative).

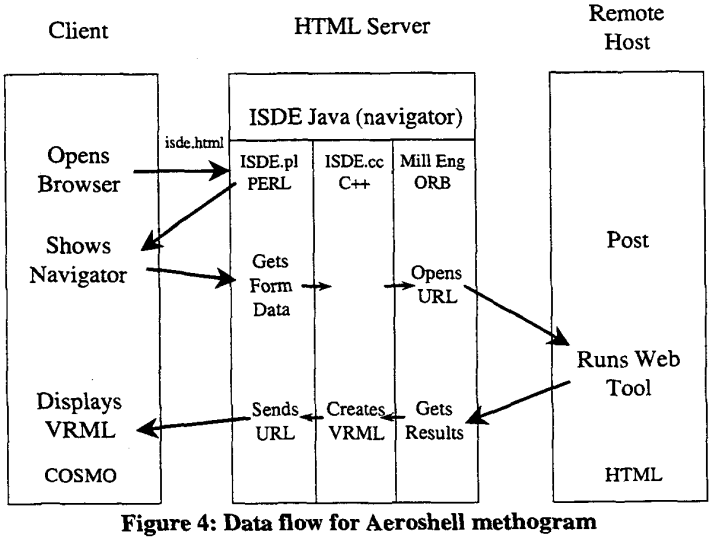


Figure 4: Data flow for Aeroshell methogram

Single User

If the user chooses "standalone", then a new copy of the Millennia Engine server is started, and no collaboration is allowed. After login, the standalone user is presented a list of projects he is authorized to view. He selects one and chooses a role from a list preselected by the creator of that project. An example role could be "browse and execute only" which allows no changes to be made other than data values. If the user is authorized to make changes, he can add or modify the methogram flow, add tools, insert attributes to methogram nodes, etc.

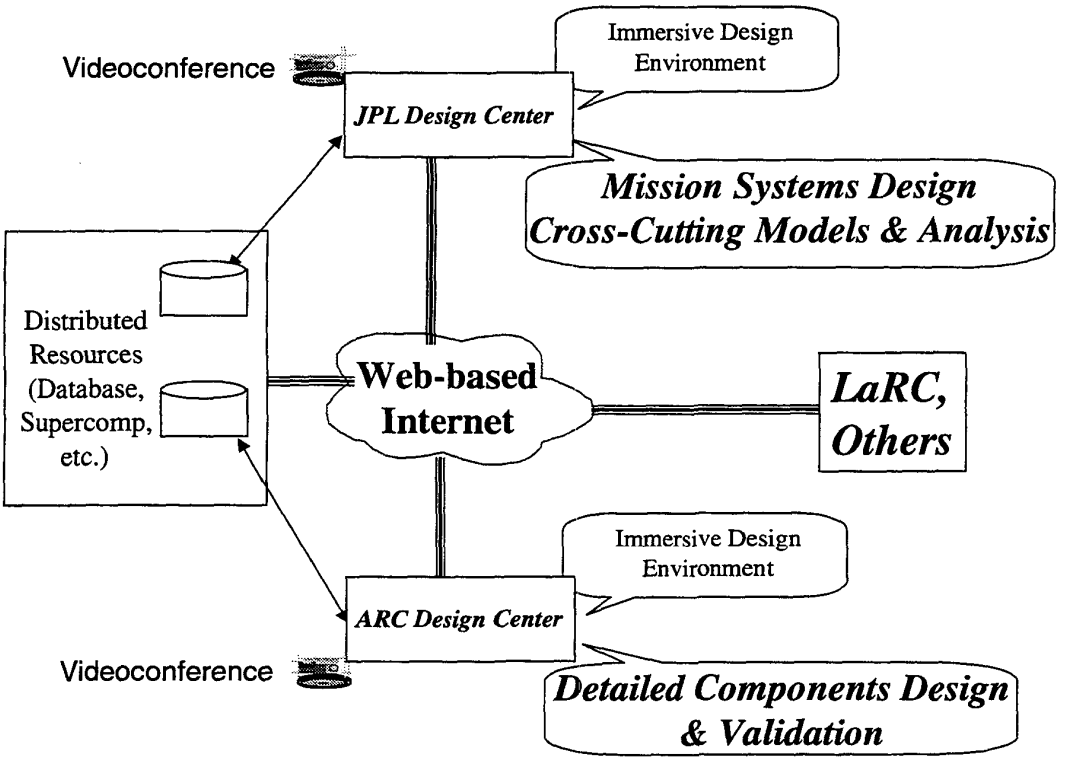


Figure 5: Collaborative Aeroshell Design between JPL, ARC and LaRC (now performed via web)

Collaborative Usage

However, if the original user login is chosen to be "collaborative", then the session broker checks whether a project selected by the user has already been opened by other collaborators, and if so, connects the new user to the already-running instance of the Millennia Engine. All collaborative users can then simultaneously browse, modify, and execute the methogram if individually authorized. Typically, modification control would be given over different parts of a methogram to different collaborators, and a system engineer could view the collaborative interaction (perhaps using standard videoconferencing capabilities in parallel with the netISDE). As each user logs off, the session broker determines whether any participants are left in the session, and if not, shuts down the instance of the Engine.

We are in the process of adding security features to the access and authorization controls described above. For example, the web servers allow secure http to enhance the privacy of their connection to the browser clients. Fine-grained role-based access control is also being considered on the methogram elements via external means.

5. CONCLUSIONS

We have described enhancement of our ISDE to allow web collaboration via Java-based browsers. This includes visualization of 3D simulations via VRML. The two reasons for this migration were a) portability of the client and b) user familiarity with, and ubiquity of, the web browser. Regarding the former, we have unfortunately found that the touted "portability" of Java is not yet a reality, since our client code (written in Java 1.1) will not run properly on browsers which do not run the "latest" update (e.g., Netscape 4.05 or later); the browser code fails with our version of Internet Explorer 4.0. We expect this issue to be resolved as the Java platform becomes more stable and mature. Regarding reason b) above, the feedback we have received from the earliest user (ARC) is that this environment will significantly enhance utilization of their computational tools (and those of others, such as LaRC). This conclusion holds despite the fact that both centers are already making the migration to web accessibility for their tools, since the ISDE provides a graphical way to create and manipulate a sequence of execution of tools from various sources. Without the ISDE, these tools could only be connected together in a fixed, clumsy, sequence (e.g., via CGI scripts, which would need to be created and maintained by programmer/analysts rather than the tool users themselves).

In parallel, we are also evaluating migration of the Virtual-Reality capabilities provided by the earlier museISDE from the original Unix platform (SGI and Solaris) to the PC platform (Windows NT). This will provide further desired

interactive 3D capabilities to the many collaborator desktop PCs. Voice recognition is also developing rapidly, though parsing and data retrieval are still weak. The stretch goal is to facilitate interactive collaboration of diverse disciplines over time and space, mediated through a PC-based shared virtual reality. We are also trying to make control of the immersive environment much more intelligent, increasingly handled by software agents (which will perhaps themselves collaborate) to assist in scenario synthesis, data analysis and presentation, database interactions, etc. We believe that fully integrated voice-activated intelligent design systems will be ubiquitous on the desktop within ten years, allowing more fully optimized designs to become realizable at a progressively earlier phase, and producing significant (and reliable) reductions in cost, risk, and time.

6. ACKNOWLEDGEMENTS

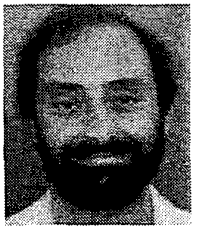
The work described was performed at the Jet Propulsion Laboratory of California Institute of Technology under contract with the National Aeronautics and Space Administration. The authors, working under John Peterson, acknowledge various contributions made by their JPL coworkers to the ISDE.

7. REFERENCES

[1] N. Lamarra and Julia Dunphy, "Interactive Sharable Design Environment for Collaborative Spacecraft Design", Proc. IEEE Aerospace conf., Snowmass, CO, Mar 1998.

8. BIOGRAPHY

Norman Lamarra leads a technical group at JPL devoted to advanced prototyping of applications utilizing emerging distributed-systems technology, including modeling and simulation, interactive graphics, distributed computing, and networking technologies. His current primary focus is on developing a collaborative virtual-reality environment to integrate spacecraft design and mission modeling. Dr. Lamarra obtained the Ph.D. degree from UCLA in System Science (Communications Systems) ('82). Prior degrees earned were M.Sc. in Electronic Engineering ('74) and B.Sc. in Mathematical Physics ('73), both at the University of Birmingham, UK. Before joining JPL in 1994, Dr. Lamarra worked for over 20 years in radar systems analysis, radar simulation, phased-array antenna analysis, and real-time multiprocessing systems.



Julia Dunphy received her Master's and Bachelor's degrees in Physics and Mathematics from Cambridge University, UK, ('63) and her doctorate in Theoretical Physics from Stanford in '67. She now works as a contractor to JPL in the areas of design research and network computing. She holds several patents and has published over two dozen papers in various areas such as magnetic recording, error-correction coding, and control of robotic vehicles (for the Mars Pathfinder Rover.

9. GLOSSARY

ARC	Ames Research Center of NASA
CGI	Common Gateway Interface (used for execution of scripts behind web servers)
CORBA	Common Object Request Broker Architecture
HTML	Hypertext Markup Language
ISDE	Interactive Sharable Design Environment
JPL	Jet Propulsion Laboratory of Caltech
LaRC	Langley Research Center
Methogram	Graphical flow of tool execution implementing a design flow
MuSE	Multidimensional User-oriented Synthetic Environment
NetMethedit	Web version of methogram editing and execution tool
NetISDE	VRML version of ISDE previously utilizing MuSE viewer
PC	Personal Computer
PERL	Practical Extraction and Report Language
SGI	Silicon Graphics Inc.
URL	Universal Resource Locator
VRML	Virtual-Reality Markup Language