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 525-3682
Pasadena, CA 91109
626-306-6148
Norm.Lamarra@jpl.nasa.gov
626-306-6153
Julia.Dunphy@jpl.nasa.gov

Abstract-- An advanced integrated environment is being developed at JPL that links collaborators who wish to perform interactive design analyses and/or mission simulations. The environment utilizes commercial technology (such as 3D visualization) where applicable, but key pieces are currently provided by software developed in-house for user interaction and spacecraft modeling. It allows a mission scenario to be built and exercised at various levels (e.g., macro or micro simulation, modeling or analysis), and integrates existing tools preferred by participants "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) are connected into a simulation which can be run from within an immersive sharable environment. 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 will discuss the problems and achievements from early utilization of this evolving interactive environment, and describe the near-term plans for shared development and deployment of the collaborative capabilities across NASA.

TABLE OF CONTENTS

1. INTRODUCTION

Problem description

In today's aerospace environment, 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 determine as early as possible which ones have highest probability of success, both before and after mission selection. The problem is therefore that the "many small low-cost missions" approach requires:

(1) successful implementation of small-scale missions with *pro-rata* reduction in cost and time for all lifecycle phases (design, build, test, operate);
(2) early determination of mission feasibility and relative cost of alternatives;
(3) rapid adaptation of proven successful missions to new objectives;
(4) rapid insertion of new technologies.

Approach

It is well known that a significant portion of the cost of developing large-scale aerospace systems is consumed by rectification of problems discovered late in the development cycle (e.g., during integration and test or even after launch). Such problems can often be traced to errors introduced early in the design, when understanding of the operational system was necessarily limited. An obvious corollary is that if such errors could be foreseen at the early design stage (e.g., during conceptual design, or pre-phase A), then significant cost/time savings and/or significant performance improvements could be achieved. We thus focused primarily on this early design phase, and attempted to enable a simplified virtual version of integration and test through simulation based on models of the mission and spacecraft subsystems. Concomitant benefits to this "virtual flight" approach are:

(1) ability to rapidly compare alternative approaches to the mission objectives;
(2) ability to re-use (and tailor) successful designs (via their models);
(3) ability to provide early insight to the Principal Investigator regarding the type, quality, and primary determining factors of the mission (e.g., science information) objectives;
(4) potential to trace the operational consequences of early design decisions, which would normally not be observable until an operational system is either built or simulated at sufficient detail;
(5) potential to track the system through development, integration and test to assist resolution of problems arising when real hardware and software are progressively integrated.

Goals

In order to implement this approach, the work described herein attempted to design an Integrated Synthetic Design Environment (ISDE) to achieve the following:

(1) ability to assemble a dynamic mission simulation from models, which may initially be imprecise and incomplete;
(2) 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);
(3) integration of analysis tools to evaluate the operational behavior of the system during such phases;
(4) visualization of key measurables such as science observations, system performance, or physical layout;
(5) ability to allow interactive modification of scenario, system, or subsystem components (such as position or performance parameters) in order to facilitate design trades;
(6) ability to reduce cost by assembling a "virtual team" without necessitating collocation.

In order to achieve these goals, we attempted integration of geographically distributed team members at two levels: first, to bring their diverse tools together within the environment; and second, to leverage their specific expertise during utilization of the resulting environment (such as described in usage scenarios below). A primary overall target was to achieve reduction of early design (through Phase A) by 80% (e.g., from months to weeks).

2. DIRECTLY RELATED WORK

This section briefly describes related work at JPL, including two internally funded re-engineering activities, called the Develop New Products (DNP) and Enterprise Information System (EIS) projects. Most of our work, however, was externally funded: by the Defense Advanced Research Project Agency (DARPA), under the Rapid Design Exploration and Optimization (RaDEO) program; and by NASA Code S, under System Integration and Test Tools. A common theme of these activities was to facilitate design improvement by collaboration of participants from widely distributed disciplines and locations. For example, DARPA is interested in large-scale development efforts that leverage technologies from commercial vendors as well as from directly-funded government agencies. NASA was similarly interested in leveraging expertise held appropriately at distinct centers, which have different responsibilities and mandates.

"Develop New Products" Project

The two-year internally-funded Develop New Products project attempted to analyze JPL's existing processes spanning the entire mission lifecycle. It then grouped these processes into several renamed ones: Mission and System Design (MSD); Develop, Build, Assemble, Test (DBAT); Validate, Integrate, Verify, Operate (VIVO); and Project Planning, Implementation, and Closure (PPIC)[1]. A key DNP architectural decision was to maximize the leverage available from commercial information technology, in order to unify and streamline these four re-engineered processes. More detailed descriptions of the processes appear elsewhere, but they are briefly described below to provide context for the ISDE work described herein. This work also follows successful prior efforts in collaboratively developing mission proposals performed by JPL's "Team X".

Team X—JPL developed this capability over the last few years in order to perform collaborative conceptual design. The team is constituted of a representative from each of several subsystem areas (e.g., Power, Propulsion, Attitude Control System, End-to-End Information System etc.), who gather in a strategic facility named the Project Design Center (PDC) to construct a concurrent-engineering model (CEM) of a conceptual design. The CEM consists of a set of linked spreadsheets whose cells contain key system performance parameters and cost values. Certain cells in the team leader's master spreadsheet are linked to those in spreadsheets on other workstations (operated by the respective subsystem representatives), via a publish/subscribe mechanism, which allows the participants

[1] Within this terminology, we are concerned in this paper primarily with the MSD process, and are attempting to assist in its development by infusion of technology (such as the integrated design environment described herein).

to modify parameters over which they have cognizance. The whole team can thus view inter-subsystem dependencies of requirements at a low fidelity, but sufficient to conduct generic performance/cost trades at some level. The entire process of synthesizing such a conceptual design can thus be achieved in a few days, and the resulting design is documented at a level appropriate for proposal submission, primarily providing an initial set of system functional requirements. More recently, the DNP project has attempted to integrate Team X into the re-engineered MSD process as follows.

MSD—The next steps in the MSD process turn the set of functional requirements produced by Team X into detailed system requirements, by integrating Mission Planning, Concept Development, and Scenario Development subprocesses. A tool (DOORS) is used to document and track the evolution of these requirements, and a new modeling tool (Foresight, from Nuthena, Inc.) allows creation of interacting functional requirements models. Dynamic operation of these models can be observed by driving them with sequences produced by the Mission Planning and Scenario Development subprocesses using other tools. This allows validation of the consistency of a set of functional requirements, as well as observation of some aspects of overall system behavior (such as total power requirement, or changing data-bus load) during a particular mission sequence. Values of key system requirements and subsystem parameters are stored, retrieved and modified in a commonly-accessible database (the Parameter Database, PDB), thus facilitating capture and management of the design trade process. Project meetings to discuss the findings observed from interactive exercise of these models can occur in a collaborative environment such as the Design Hub (DHUB), which collocates subsystem developers and their tool suites with common resources. Attempts to resolve conflicts between the subsystem elements can be made by dialog between the System Engineer and groups of contending subsystem engineers.

DBAT—The lifecycle process then moves from the functional into the physical domain through the detailed design and development phases (B, C, D), during which subsystem physical designs are developed using tools preferred by each electronic and mechanical discipline (e.g., ILOGIX, Mentor Graphics, Cadence, ProE). Parts of this process can also be performed collaboratively in the DHUB environment, and evolution of the system design can be captured in ever-increasing detail in the PDB, including physical models of subsystems and their behavior. It is currently not considered feasible to construct a simulation of the complete system from synthesis of these detailed subsystem models, due to technological and economic limitations.

VIVO—However, when detailed behavior models of particular subsystems exist, or in some cases prototype hardware exists, a "live" (partial) system model can then be constructed in the Flight-Systems Testbed (FST). This requires system integration to be performed between whatever subsystem representations exist (real or modeled), and the resulting hybrid system can be driven by test sequences to observe behavior or analyze instrument performance. Results and measurements could then be captured in the PDB and compared to the system requirements. This would allow feedback to the MSD phase, but now with an increasing fidelity which progressively represents more of the complete operational spacecraft.

PPIC—This process provides management of the project and interactions between the above processes and their subprocesses. In particular, considerations of integrated cost, schedule, workforce and risk are performed in this process via management interfaces to the above processes and to the larger JPL business environment (e.g., workforce skill level and resource availability).

EIS Project

The internally-funded two-year Enterprise Information System (EIS) project analyzed JPL's requirements for a lab-wide Information-System Architecture based on industry standards [1]. It is currently in the process of implementing and integrating several recommended infrastructure services (e.g., file, network, data access, messaging, system management, security, and directory) and operationally deploying them. This includes engineering and staffing them for continuous operation, customer training, and 24x7 support. The EIS is intended to facilitate seamless interoperability among JPL processes (such as MSD, DBAT, VIVO, PPIC), resources (such as the PDB), and facilities (such as the DHUB, PDC, FST). Three of the most mature and widely used lab-wide services provided by EIS are: a secure, distributed, redundant file system (~300GB of RAID storage utilizing Transarc AFS/DFS); email (currently about 15,000 inbound messages per day); and the intranet (which connects about 14,000 Ethernet nodes at 10 and 100 Mbps). These services are, in fact, global since JPL's intranet extends to a mission-critical extranet connecting the Deep Space Communications Centers on three continents for 24x7 Deep Space Operations.

A major goal of the EIS is to integrate enterprise-scale global services such as security, file, and directory. These are currently based on Transarc's DCE/DFS, a commercial implementation of the Open Group's Distributed Computing Environment (DCE) and Distributed File System (DFS). As these services mature and achieve widespread adoption, the further goal is to leverage them to assist in integration of JPL's business processes. An example might be rapidly providing a new project with an robust, standard, integrated information environment and tailored tools that allow the project to begin doing real work from Day 1 and support its entire lifespan.

3. THE INTERACTIVE SHARABLE DESIGN ENVIRONMENT

Overview

An immersive design environment is being developed to allow designers to perform important pieces of the much larger processes described in brief above, but at a much earlier stage than presently possible (even in the emerging DNP architecture). The ISDE enables integration of functional-requirements models with physics-based models (e.g., of instruments and real-world phenomena [2]), in order to allow interactive design based on observation of spacecraft performance in a simulated mission context.

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. State-of-the-art tools are used when possible, but sometimes only best-practice ones are readily available. The environment also allows experimental or "home-grown" tools to be used, e.g., probabilistic analysis methods.

The developers and early users of the ISDE are thus benchmarking existing tools for inclusion in the new process as it emerges, and can hence provide clearer definition (e.g., to commercial suppliers) of enhancements which would be required to make development of next-millennium spacecraft progressively more seamless and adaptable.

Components of the ISDE

The components of the ISDE are pictured schematically in Figure 1, which also shows current participation of various NASA centers and vendors.

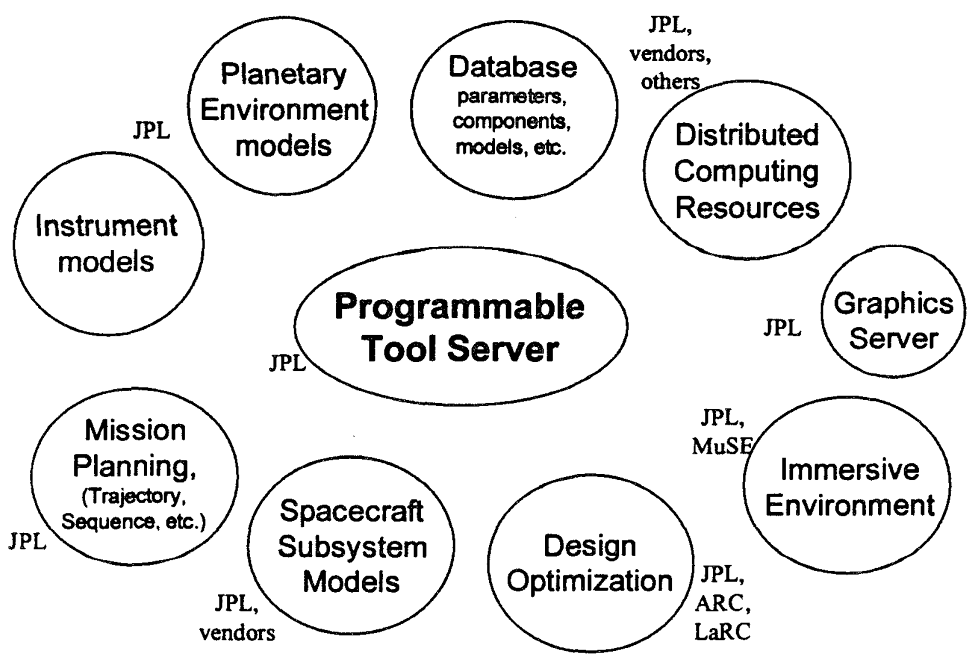

Figure 1: Schematic Representation of Immersive Synthetic Design Environment (ISDE) Components

MIDAS and the Millennia Engine

The Millennia Engine provides an infrastructure to integrate design and analysis tools and present the user with a "plug and play" interface that allows many of the above goals to be met. It can provide access to a database of components, analysis tools, visualization tools, drawings, and documents. A designer can develop a reproducible design methodology by first connecting these resources together and then performing interactive exploration of the operation of the designed object in appropriate context (i.e., seamlessly moving back and forth between simulation and design). The methodology is generated and modified graphically (*methogram*), and can allow capture of the process that the particular designer is accustomed to following when making decisions about the form and attributes of each component. The methogram is saved in the database and can thus be reused either in another part of the design or adapted for a later design.

Obvious candidates for transfer into methograms are design processes which are quite straightforward and repeated for each new design. For example, a certain shape may be chosen with beginning dimensions, then a thermal analysis may be run, and a new dimension chosen based on the results of the analysis. The process would be repeated until satisfactory results are achieved. All of these steps can be described in a methogram that can be modified or reused.

After synthesizing a methogram, it is debugged (via MIDAS graphical facilities) until it is as general and detailed as possible. The user, who may or may not be the originator of the *methogram*, can then provide the input, in the form of design requirements, and iterate the methodology to arrive at a point design. If the requirements change (as they often do) the user can repeat the process in a matter of minutes instead of having to manually execute the design steps. The Millennia Engine (MIDAS without the user interface) thus allows capture of a key designer's expertise in a dataflow design graph. This is not a "knowledge-based" approach in the usual sense, but is rather a prescriptive representation of the methodology that an expert would use in designing some aspect of the spacecraft. The *methogram* is then usable by a generalist who is then able to perform more integrated conceptual design and trades with a very small team, thus enhancing the MSD process described above).

We have completed the first phase of our effort to make design and analysis tools more accessible and collaborative. Capture of the design process in an electronic form also paves the way for computer-aided optimization, allowing vastly greater search of the possible design space and interaction with the requirements. An almost arbitrarily complex set of methograms can be executed on a network of workstations, including supercomputers at JPL or other collaborators' facilities.

Human Interface

The Millennia Engine (developed over the last few years) provides the necessary capability in linking components and analysis tools in order to create and evaluate a design, but integration with the ISDE (developed more recently) enhances the capabilities of the human user to create and interact with a mission simulation. We are using a commercial software platform for development of immersive applications (MuSE Technology, Inc.) to allow the mission to be visualized (e.g., the spacecraft, a target planetary body, and its environment). The MuSE system is connected to the Millennia Engine through a CORBA-compliant interface which we developed using Orbix (IONA Technologies). The virtual-reality "front end" of the ISDE thus allows a user to explore a design immersively in the mission context. For example, the user can request (via voice command) that a certain analysis tool be connected to a component of the design (via the methogram, which is then executed), and the results are presented to the designer using advanced visualization techniques. Such techniques include color (e.g., representing temperature or stress on the surface of a physical component), numerical graphics (e.g., a performance plot posted in the virtual world), touch (e.g., used for navigation in the virtual world), and various sound cues. Such data presentation methods have been shown to be capable of increasing the efficiency of a designer by up to 100 times over standard flat screen displays in traditional tools.

The technology employed in the ISDE is that of persistent objects which can be passed between system elements (e.g., computers) using CORBA, as shown in Figure 2.

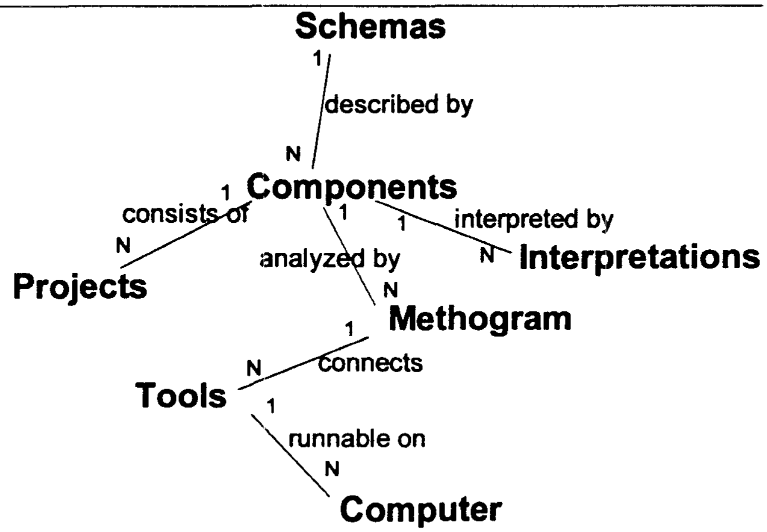

Figure 2: Relationships Between Data Elements

In this drawing, a line with N at one end and 1 at the other means that there may be N objects of that type which are related to 1 at the other end. So, for example, the lowest line shows that there may be N computers on which a particular tool can be run. The line above that says that a methogram may require N tools to be run simultaneously or sequentially. Within the hardware components which make up the spacecraft design there are relations which are also managed by the database. An example of a hierarchical set of such relations is shown in Figure 3 for a spacecraft conceptual model.

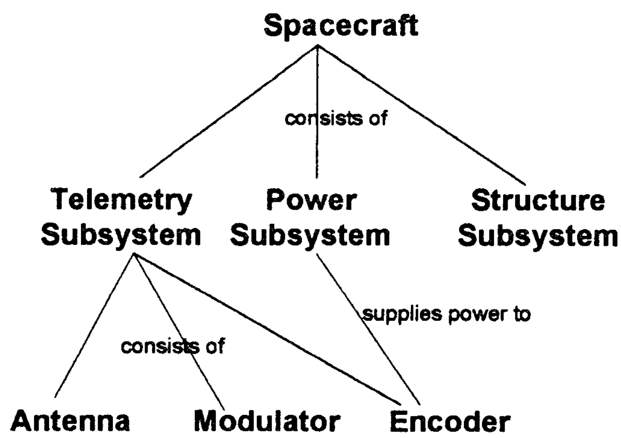

Fig. 3 Relationships Between Modeled Hardware Components

Database Interface

The database provides a CORBA-compatible repository for all these objects. JPL has already defined CORBA interface objects for some of these including Projects, Schemas, Methograms, Tools, Computers. At present, these objects are part of the Millennia Engine internal database, but we are in the process of transferring them to the design database.

Overall Architecture

Based on the above discussion, the overall software architecture for the MuSE-based Integrated Product Design tool looks like Figure 4. The MuSE system is connected via its shared-memory interface to the ISDE Gateway, which is a CORBA client of the Millennia Engine.

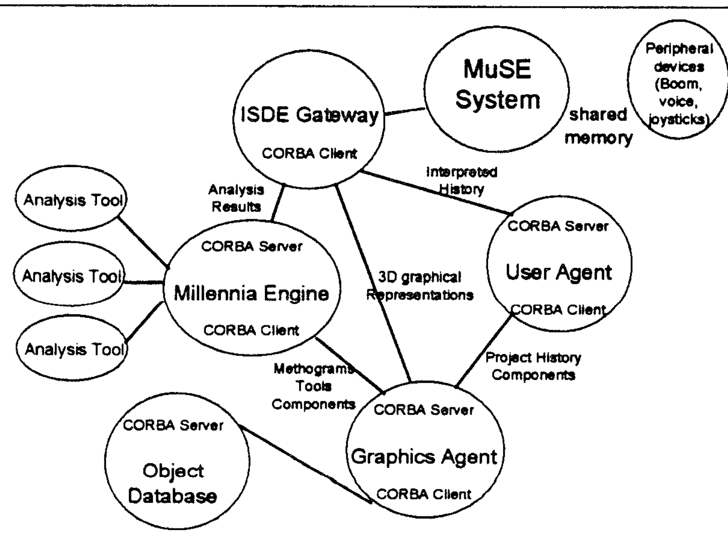

Figure 4: ISDE Software Architecture

While the above architecture represents the ISDE system configuration, many other specialized architectures are possible, such as simple traditional MIDAS configurations.

Sharable Environment

A key advantage provided by the use of a commercial virtual-reality system is exemplified by the most recent enhancement of the MuSE software: namely the ability for several participants to interact simultaneously with the simulation from remote locations via the network. This capability is called "Continuum" by the vendor, and is not merely a "remote display" capability, but actually truly independent views of a distributed simulation. Participants can either be synchronized or choose to change or stop "local" time. They can share the same perspective or individually navigate the virtual world, perhaps to view components of more local interest. Each participant can see a representation of the others on the display, allowing perspective of their inter-relationships, and the provided voice interaction allows discussion of components or findings (e.g., "if you move around your subassembly to your left, you can see that my subassembly is overheated, inaccessible, etc.").

User-Interface Agent

To assist the human designer working from within the ISDE, we are developing a "virtual human" assistant. Designers are often senior employees who are unfamiliar with Virtual Reality environments. We hope that the virtual human will be lifelike enough to facilitate their use of the ISDE. We are using the "Jack" product from the University of Pennsylvania for our virtual human. Jack will offer advice about how to use the equipment when requested by the user. He will also be able to undertake tasks such as search for a particular methogram or standard component when requested. As a first step, we have demonstrated control of the Millennia Engine by natural language command (described below). The user thus already appears to be conversing with an agent inside the virtual world.

Natural Language Parsing

There are two ways commonly used to teach people a foreign language. One is to teach them all the parts of speech—nouns, verbs etc.—and the rules of grammar. The other is called the "natural language approach". In the natural language method, people learn languages by learning whole phrases and discovering what they mean by asking questions of the teacher in phrases they already understand. This is the way children learn from their parents. We are trying a natural language approach to teaching the ISDE to understand the verbal commands of the operator. The ISDE is given a set of basic commands using a schema like the following:

SEMANTICS: 1 <Load Project $0>
SYNTAX: <load project %s> $0
SYNTAX <get project %s> $0
RESPONSE: <project %s is loaded> $0
ADMONITION <project %s does not exist>

The SEMANTICS line is the actual instruction sent to the engine and can consist of a series of parametrized CORBA calls. The lines designated SYNTAX represent the present knowledge that the ISDE has concerning the alternative ways the operator might ask it to perform the operation. The parameters are designated as $0, $1 etc. The RESPONSE line represents the generated voice response if the operation is successful, while the ADMONITION is the response when there is an error.

If the user says "find project Neptune for me" rather than "Load project Neptune", the agent will reply that it doesn't understand. The user will then rephrase until he hits the correct phrase, at which time the agent will ask if "find project Neptune for me" means "load project Neptune". If the user agrees, the ISDE will add this paraphrase as a syntactic alternative and will understand it in future. Of course, other constraints need to be applied in practice, but the system is already working quite well. We have identified about 20 "primitive" operations commonly performed during a design session. These include selecting a component, selecting an attribute of the component and then retrieving or setting the value of the attribute. While some attributes are specific to a component, others such as position and dimensions are universal. A statement like "Move the thruster 2 cm" is translated into "increment the attribute 'position' by 2 cm". Since there is such a small set of primitive operations, and there are very few legal combinations, we believe that the total number of useful phrases is unlikely to exceed 100;a large semantic database is thus unlikely to be generated.

Graphics Agent

The Millennia Engine achieves display of graphical components, such as a spacecraft assembly or planetary body, via a graphics agent. This is actually a server that provides an intelligent interface to the database of images, via a CORBA interface which can be used by the ISDE just as that for the Millennia Engine is used. For example, if the methogram is executing an orbital maneuver during a simulation of part of the mission sequence, then it "reports" the position and attitude of the spacecraft as a list of graphic components to the ISDE. This list is then given to the graphics agent, which finds them from the database and converts them into a representation that can be directly rendered in the MuSE environment from the appropriate perspective. This keeps much of the application-specific graphical intelligence (such as color applied to represent temperature) outside the MuSE application, which thus only needs to be concerned with performing the more generic operations such as local screen updates and handling of user interactions.

Optimization Agent

Methograms in the Millennia Engine can automate algorithms; however, in order to assist the user in intelligent parameter optimization, control of the execution flow and parameter values can be given to an external optimization agent, which is currently being integrated. This work is focused on developing a reconfigurable genetic optimization system [4-7], which generates candidate genetic algorithm configurations and optimizes an objective function given a high-level description of the problem. This has been demonstrated on a Mars Microprobe penetrator; recently, an extension (incremental evolution) has been shown to result in significant improvement of the optimization performance. Hypothesis-testing algorithms are also being investigated for efficient evaluation of candidate spacecraft designs. This enables significantly more efficient evaluation of candidate designs than previous methods (e.g., order of magnitude improvement in efficiency over existing statistical techniques [5]).

Physical Implementation of ISDE

Currently, the ISDE application runs on both Sun (Creator3D) and Silicon Graphics (e.g., Octane) workstations, since the MuSE development environment supports both these platforms. An additional high-end SGI configuration is available, which possesses several additional (though not essential) peripherals: an Infinite Reality Graphics Engine driving a quad PowerWall display; head-mounted boom display; 3D LCD glasses (Crystal Eyes); fly box; sound synthesis and (PC-based) voice recognition equipment. The Millennia Engine and Graphics Agent run as CORBA servers on a low-end Solaris SPARCstation, and other applications can be run by the Millennia Engine via Unix remote shell or via PVM on heterogeneous platforms (PC, Mac, Sun, Cray, etc.).

4. EXAMPLE USER SCENARIOS

The present state of the design of the ISDE system is that all basic functions are available. What is lacking is a good set of methograms to use during the design process. Methograms need to be designed by hardware specialists who are familiar with the detailed process of analyzing the design, and this is out of the scope of software developers like us. Getting a specialist (e.g. an antenna designer) to encapsulate their design methodology in a methogram has proved daunting, because the immediate payback is not obvious to them. We believe there needs to be an institutional commitment to parametric design techniques in order to achieve real progress.

In 1995, a group of spacecraft design experts was employed to use the precursor of the ISDE, the MIDAS system, to generate methograms and use them to design a Mars orbiter similar to the actual Mars 98 design. They concluded that there was an 80% reduction in the effort for each new spacecraft design, building on the one-time cost of producing the methograms. Cost Modeling was included in this effort by attaching a cost to each component and an assembly cost to each methogram. While this was crude, it showed that such a cost model could be built. A better approach would be to hide such a calculation and keep a tally of the total cost of the design in a separate part of the screen.

Single User

A design scenario for the year 2000 might consist of a spacecraft designer sitting at a control boom and looking at a spacecraft in orbit around the Earth. With a voice command, the designer can stop time and point the spacecraft to the star that it is observing. The designer can then request that a calculation of the light path through the optical system be performed using the JPL I-MOS software. The results would return in the form of twin light paths becoming visible on the spacecraft image. The designer could then say "Remove panel A" and this would then show the light path interior to the spacecraft. He could then make some modification (e.g., move mirror position) and redo the calculation, all with voice commands. Then he could ask "Jack" why the mirror position was so close to the telemetry and Jack will search the data base and reply verbally by repeating conversations that were exchanged on the subject back in 1998. All of the visual elements in this scenario need to be created ("authored") using an immersive design authoring tool. We are presently evaluating a tool kit from Virtual Prototypes Inc. We believe that their object-oriented data-driven approach and their animation methodology (using direct tie-in of data from simulations into scene parameters) may work well for the ISDE. However, integrating this into the MuSE environment will present a challenge.

Collaborative Usage

A future collaborative scenario could similarly be envisioned as follows: a designer enters the ISDE, restores her previous aeroshell design session and reviews the results. She then determines that she needs to bring in a remote collaborator, who is then electronically connected to the environment. He first sees her perspective, then moves to another view more pertinent to his area of expertise. Together, they discuss modifications to the design, making changes to design parameters, component locations, or the scenario, or replacing design components with alternatives. They then decide to run a static thermal analysis, which they visualize immediately as a colored temperature distribution over the aeroshell. When this is satisfactory, they decide to rerun the descent portion of the mission interactively, simultaneously viewing the simulation from their own viewpoints (e.g., one looking at aeroshell temperature distribution, the other looking at the trajectory from the flight control perspective). During and after the simulation run, their discussion is recorded along with their findings. This can be used to annotate the design history for later search and retrieval as described in the previous scenario. Figure 5 shows a currently-funded collaborative activity between JPL and two other NASA Research Centers (Ames and Langley), aimed at providing the first example of just such a real-time ISDE-mediated collaboration for design of a precision lander for the Mars program.

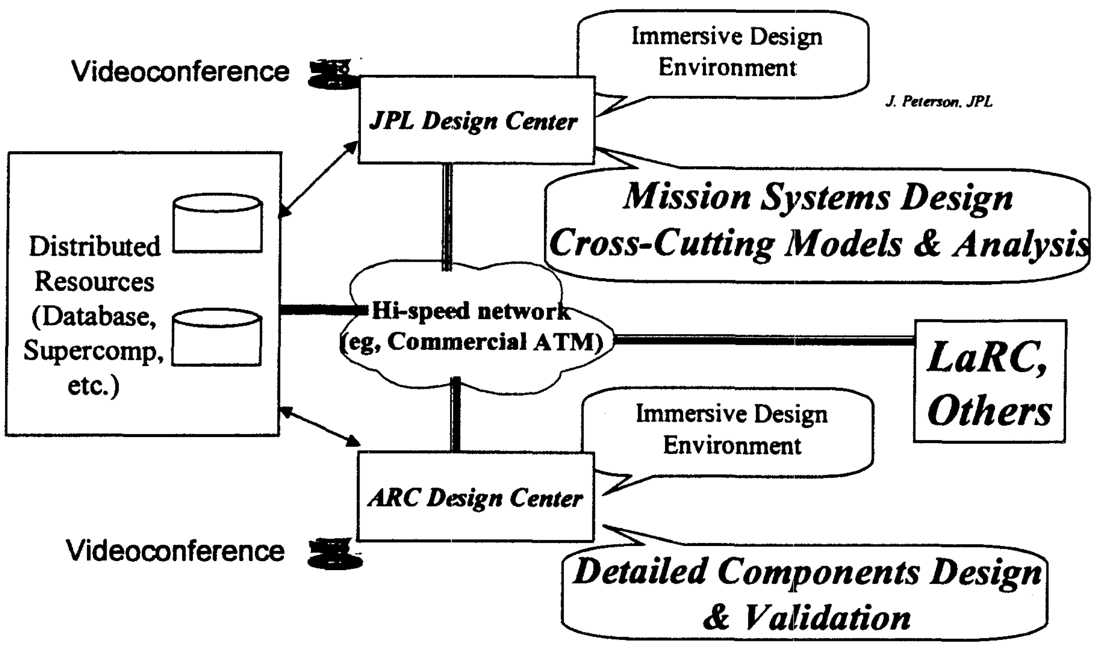

Figure 5: NASA Collaborative Design Applicable to a Mars PrecisionLander (JPL, ARC, LaRC)

5. LIMITATIONS AND TECHNOLOGY GAPS

The existing ISDE is incomplete and is currently in experimental form. It is difficult to reconfigure, requiring some custom work by the developers to integrate new tools or models, and modifications to the graphical user environment requested by users can be particularly challenging. However, all of the features mentioned above have already been demonstrated to some level, and plans exist for evolution towards the robust, deployable, maintainable environment which would be required before widespread and long-term customer acceptance can be expected. We have identified short-term customers who are sufficiently interested in the potential of the existing features both to support continued development and to assist in providing metrics by which the usefulness and completeness of the evolving environment can be judged. Such user support and feedback is essential before committing significantly greater funding to this activity, and will determine the user-driven priorities for feature-set implementation.

Although some aspects of Mission Planning can be accomplished using the tool, it is not integrated with preferred planning tools at JPL. Again, the relevant methograms would also need to be generated. Generic Scenario Building has not yet been addressed, though there are several tools used at JPL which might be adapted to assist in this. We have generated a detailed user guide for the MIDAS functions of the ISDE, but are still working on documentation of the graphical features and commanding system. For methogram production, online help is available. Configuration Management is currently quite rudimentary (methogram versioning).

There are several areas in which current technology lags the requirements for such User Scenarios. Commercially-available frameworks and tools to replace the home-grown varieties in current use currently appear less capable than the ISDE. Significantly better integration with database technology is also required, particularly with intelligent databases to perform associative search and natural-language query for text and non-text objects. A robust scenario builder could enhance (or even replace) the

Millennia Engine; as higher-speed network and computing infrastructure evolves, they will progressively allow higher fidelity to be obtained from simulations of such scenarios when required. Eventually, functional models of components could be replaced with physical models at the appropriate fidelity, easing the transition from the virtual world into the "real" world of hardware and flight testbeds.

6. CONCLUSIONS

The immersive design center is comingsoon[3]. Many of the required pieces are now generally available and computer and graphics technology are fast enough to support them. Voice recognition is also developing rapidly, though parsing and data retrievalare still weak. The stretch goal is to facilitate interactive collaboration of diverse disciplines over time and space, mediated through a shared virtual reality. We expect this soon to be hosted on a PC platform which exceeds current workstation performance, but which will soon be the "standard inexpensive desktop". Control of the immersive environment will also become 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 thus believe that fully integrated voice-activated intelligent design systems will be ubiquitous 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.

7. ACKNOWLEDGEMENTS

The work described, led by John Peterson, was performed at JPL, California Institute of Technology under contract with NASA; the authors acknowledge the various contributions made by their JPL coworkers to the ISDE. Jose Salcedo assisted in Muse/Millennia development; Ansel Teng in modeling of instrument and planetary environments; Al Fogel integrated Foresight models;Kaly Rengarajan worked on database integration; Imin Lin provided architectural design for software development and integration; Alex Fukunaga worked on integrating intelligent optimization; Bob Glaser provided expertise in the mechanical design area; and Celeste Satter provided system-engineering and connections to JPL mission personnel. DaveOlynick led the collaborative activity with ARC/LaRC. John Azzolini provided input from the GSFC related programs.

8. REFERENCES

[1] J.S. Jenkins et al., "Enterprise Information System Architecture Document", JPL internal D12991, Dec. 1996

[2] M. Lee, R. Swartz, A. Teng, and R. Weidner, "Encounter Geometry and Size Data Gathering Simulation", Proc. AIAA Conf. On Guidance and Control, Aug. 1997.

[3] A.K. Noor, S.L. Venneri, J.M. Housner, and J. Peterson, "A Virtual Environment for Intelligent Design", Aerospace America, April 1997 pp 28-35.

[4] Fukunaga A, Chien S, Mutz D, Sherwood R, Stechert A. "Automating the Process of Optimization in Spacecraft Design", Proc. IEEE AerospaceConf., Snowmass CO, Feb. 1997.

[5] Fukunaga A, "Application of an Incremental Evolution Technique to Spacecraft Design Optimization",Proc. IEEE Int. Conf. on Evolutionary Computation, April, 1997.

[6] Fukunaga A, Stechert A. An Evolutionary Optimization System for Spacecraft Design. Proceedings of Tenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, June, 1997.

[7] Fukunaga A, Stechert A., "Towards a Self-configuring Optimization System for Spacecraft Design", Proc. Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, July, 1997.

9. 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 herMaster'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.