Virtual Mission Systems for Multi-Disciplinary Engineering System Design

Meemong Lee* and Richard Weidner†
Jet Propulsion Laboratory, Pasadena, CA 91109

To ensure the science return while minimizing the cost and risk of a mission, a lifecycle-wide concurrent engineering design process must be established. This paper describes the Virtual Mission Operation Framework (VMOF), which provides distributed modeling, simulation, and visualization services. The goal of the VMOF is to enable concurrent design of a mission system and operation of that system by providing operational perspective to the system designers and system perspective to the operation designers. Concurrent development of the both perspectives requires interdisciplinary modeling and simulation of the mission-operation activities and the system architecture. The VMOF focuses on tracking and resolving interdependencies among the engineering processes and the subsystems while maintaining the compatibilities with discipline-centric operation activities and system architecture.

I. Introduction

A space science mission starts with a set of science questions about natural phenomena, and it transforms into specific measurement objectives and science-return requirements. The measurement objectives and science-return requirements drive the requirements for the mission operation, spacecraft system architecture, and instrument systems. To ensure the desired quantity and quality of the science data products while minimizing the cost and risk, a lifecycle-wide model-based engineering process that can easily adapt to a mission-specific science traceability matrix must be established. The lifecycle-wide model-based engineering process must also provide concurrent and collaborative system engineering among the instrument system, the spacecraft system, and the mission operation [1].

Mission lifecycle consists of three major phases, formulation, implementation, and operation. During the formulation phase, system-level designs are developed by performing various trade analyses among a wide range of options to ensure the optimality of the design with respect to the mission objectives and project resources. During the implementation phase, the subsystems are designed, developed and verified against the system-level design. The subsystems are then integrated and tested for operation readiness. During the operation phase, the flight system is launched and the spacecraft is navigated, controlled, and monitored toward the science targets. After a successful encounter, various telemetry data are collected from the science payload systems, engineering and science data products are developed, and finally scientific discoveries are shared with the general public.

In order for a mission to be successful, everyone must work together toward the same mission objectives and clearly understand his/her role in achieving those objectives. The challenges in providing a lifecycle-wide continuity to all mission teams are enormous due to the typical long duration of a mission lifecycle, the multiple engineering teams involved in each lifecycle phase, the multiple engineering disciplines involved in each engineering team, and so on. The transition of engineering teams from one phase to the next introduces many technical and managerial challenges. In particular, disconnects between the design team and the operation team have often been major drivers for high operation cost and mission risk. How can a mission

1. Share accurate understanding of operational objectives among all mission teams?
2. Validate operability of a system before it is built?
3. Verify science return before receiving telemetry?

These are some of the important questions that have been addressed by the system engineering community at Jet Propulsion Laboratory.

---

* Section Staff, Modeling and Data Management Section, MS168-315.
† Technical Group Supervisor, Mission Simulation and Instrument Modeling Group, MS168-315.

The Virtual Mission Operation Framework (VMOF) is one of the project lifecycle engineering process-improvement efforts at JPL. The goal of VMOF is to enable concurrent design of a mission system and operation of that system by providing operational perspective to the system designers and system perspective to the operation designers. Concurrent development of the both perspectives requires interdisciplinary modeling and simulation of the mission-operation activities and the system architecture. The VMOF focuses on tracking and resolving interdependencies among the engineering processes and the subsystems while maintaining the compatibilities with discipline-centric operation activities and system architecture. As shown in Figure 1, the VMOF interfaces with the design activities, design environments, and the subsystem discipline areas via three types of frameworks, model service framework, virtual simulation framework, and distributed visualization framework.

Figure 1. Collaborative Design Engineering Frameworks

The three service frameworks of the VMOF facilitate collaborative engineering design in three areas: 1) distributed model service framework for multi-disciplinary requirements validation and verification; 2) virtual simulation service framework for system-level operability validation; and 3) distributed visualization framework for mission objective verification. The model service framework provides design-process and product-interface mechanisms, develops heterogeneous model integration protocols, and supports virtual simulation framework with integrated system models. The virtual simulation service framework provides virtual prototyping of the integrated system models, activity and command definitions for operation scenario development, and telemetry-state definitions for operation behavior logging. The distributed visualization service framework integrates graphical models, graphical states, and viewing geometry in real-time. The visualization service framework interfaces with the other two frameworks for mission models and mission states. The above three frameworks internally operate numerous model types. Figure 2 captures the model types involved in traditional modeling, simulation, and visualization, and their relationships. The technical approaches and implementation details of each framework are described in Sections II, III, and IV. Current application areas of the VMOF and future directions are discussed in Section V.

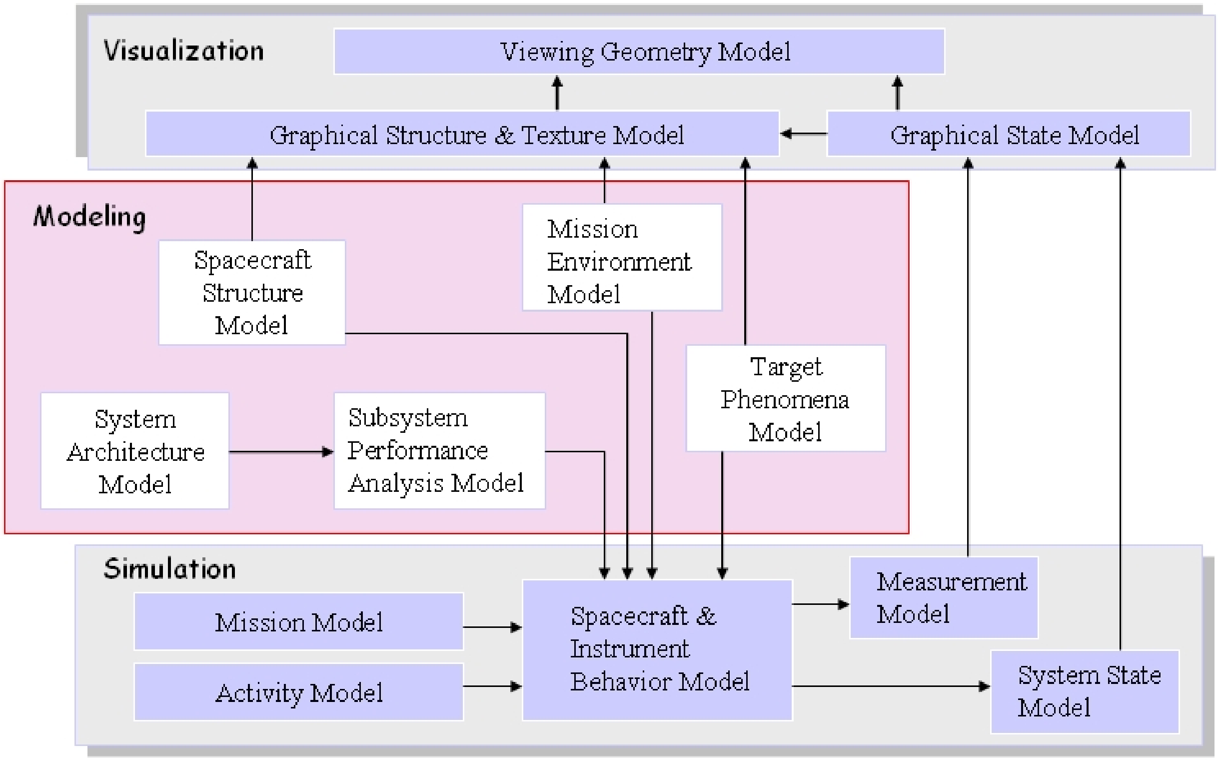

Figure 2. Modeling, Simulation, and Visualization relationship

II. Distributed Model Service Framework

A model is a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process. An engineering model in this paper is defined to be a software representation of a model of an engineering system. The software representation can take many forms depending on the type of the engineering system and the property of the engineering system that is being described. Each subsystem engineering discipline employs numerous modeling tools and develops a wide range of models to analyze the performance of the subsystem. However, these engineering models are not adequate for multi-disciplinary trade analysis during the formulation phase because they lack inter-operability and cannot handle the design uncertainties. Traditionally, low-fidelity engineering models are created during the formulation phase for system-level trade analysis. Unfortunately, the system-level models often are not compatible with the subsystem-level models; thus they are discarded during the implementation phase.

The major challenges of the model service framework are threefold. First, how to organize the formulation phase engineering models so that they can be aligned with the design processes and design products? Second, how to integrate heterogeneous model types so that operability and science return can be rapidly and comprehensively verified? And finally, how to implement formulation-phase engineering models so that they can be progressively replaced by the implementation-phase engineering models? The distributed model service framework meets these challenges by implementing various interface mechanisms that are aligned with the design process, simulation framework, and subsystem analysis tools.

A. Design Process Interface [2]

The distributed model service framework provides two independent interfaces, one for the system-design process and the other for the operation-design process so that the system-centric design and the operation-centric design can be concurrently developed and iteratively traded. Currently, the system-design information is organized under two main areas, a spacecraft system area and a payload system area. The spacecraft system area is divided into navigation, attitude and articulation control, power, and telecom subsystems and the payload system area is divided into various instrument types including camera, radar, and spectrometer. The operation-design information is also organized under two main areas, a mission area and a science area. The mission area defines mission duration, trajectory, and mission-level resource constraints. The science area defines target coverage, resolution, and downlink frequencies.

The design process interface mechanisms employ a set of Excel-based design-parameter templates that can be dynamically populated to describe a specific mission. A subsystem may employ one or more templates to organize the design products. The parameter templates are connected to a distributed design product database so that the design parameters can be stored and retrieved readily. A project workbook may contain multiple antennas and multiple cameras where the antenna parameters and camera parameters are loaded from the design product database via the telecom template and the camera template as described in Figure 3. The templates can be evolved while maintaining the backward compatibility by providing a version control.

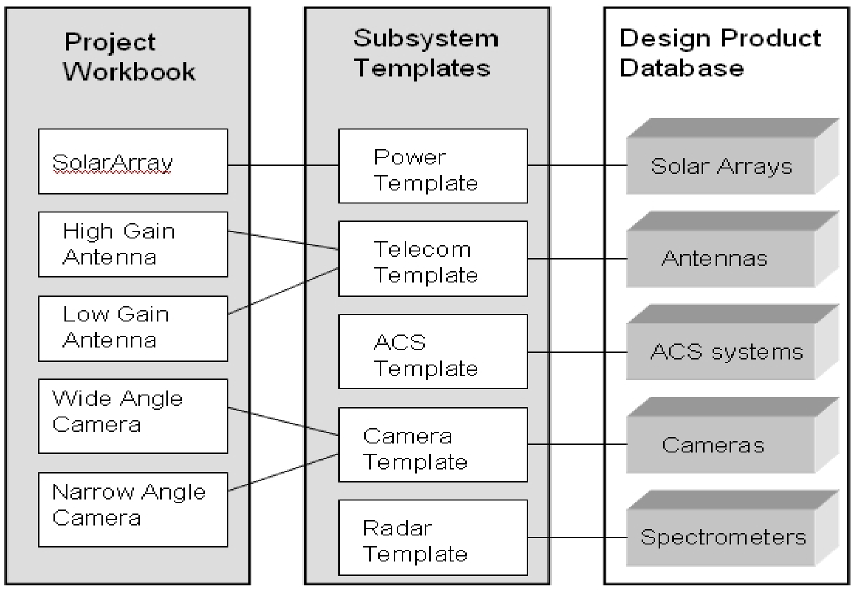

Figure 3. Subsystem Templates and Design Product Database

B. Virtual Simulation Framework Interface [4]

The VMOF supports concurrent engineering processes between the system design and operation design. This is depicted in Figure 4 where the system view addresses performance and behavior analysis of the world *(in this context, the world includes the geometric, radiometric, and dynamic properties of the physical universe that are relevant to the mission*), the spacecraft system, and the science payload while the operation view addresses interactions between them during a mission. Currently, the system view overrides the operation view during the formulation phase, and this impacts the operation phase negatively. The simulation framework enables the operation view generation during early design. The simulation framework interface mechanisms employ a set of mission information agents to enable virtual prototyping of the world, the spacecraft system, and the science payload. The mission information agents handle non-parametric mission information, such as trajectory, structure, radiation field, and target phenomena. Each mission information agent is capable of understanding and interacting with its environment on the behalf of its user and forming and executing rudimentary decisions.

The mission information agents are a subset of intelligent mission model agents employed by the VMOF. The combination of search capability with some intelligence in software "agents" has been researched for years [5,6]. An intelligent software agent is a virtual person that represents the user in finding the information the user wants without requiring the user to personally wade through all of the web pages. An intelligent model agent, in this paper, is a proxy model that represents the application in finding one or more real models in remote locations. The use of proxy models is discussed below.

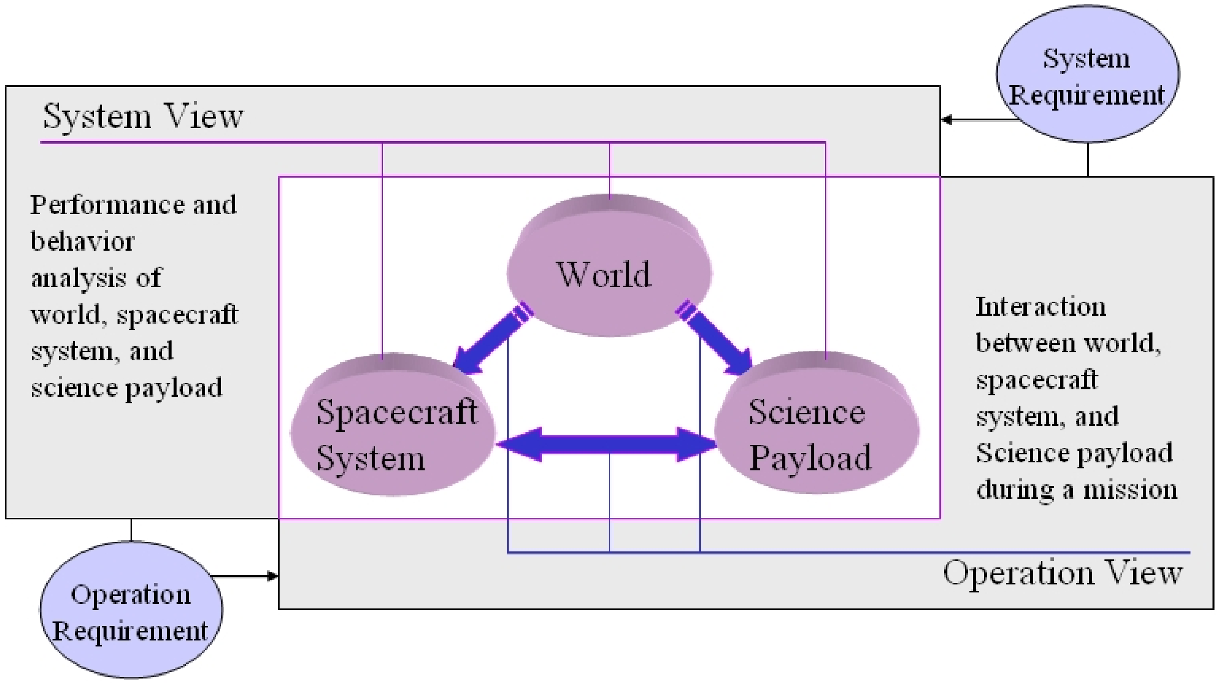

Figure 4 System View and Operation View Interactions

C. Subsystem Discipline Mode Interface

The VMOF constructs the formulation-phase engineering models as an extension of the implementation phase engineering models as much as possible. The extension is performed by the discipline engineers to accommodate the unknowns, uncertainties, and options. In order to cope with the lifecycle phase-wide variations, more than one engineering model may be required for a subsystem. When multiple models are involved to represent a subsystem as the lifecycle phase progresses, it is difficult to handle the technical inconsistencies across the models. The VMOF employs one proxy model per subsystem and per lifecycle phase and the proxy model interacts with multiple legacy models as made available by the respective disciplines.

Each proxy model provides a set of interface protocols so that complex engineering applications can be developed involving multiple legacy discipline models without being constrained by their implementation inconsistencies. A proxy model may be supported by a service broker and a model master in order to handle multiple legacy models. For example, the proxy model for the telecom subsystem is supported by two types of legacy models, TPF (Telecom Predictor and Forecaster) for deep space missions and STK (Satellite Took Kit) for Earth orbiting missions. The TPF and STK are activated by a telecom master model which prepares analysis requests and performs necessary transformations to the analysis results. The service broker establishes client connection and passes client request messages employing a data structure independent from the subsystem analysis models and the applications.

III. Virtual Simulation Service Framework

During the operation phase, the ground system commands the spacecraft system, receives telemetry data, monitors the health state of the spacecraft system, and analyzes scientific measurements from the science payloads. Figure 5 describes the operation phase interactions involving a flight system and a ground system where the flight system is composed of a world, spacecraft, and science payload and the ground system is composed of operation planning, telemetry processing, and spacecraft and instrument data analysis. The Virtual Simulation Framework provides three functions to enable comprehensive mission operation simulation: 1) Virtual prototyping of the world, spacecraft, and science payload, 2) operability validation and operation planning, 3) time-based spacecraft operation simulation and telemetry generation.

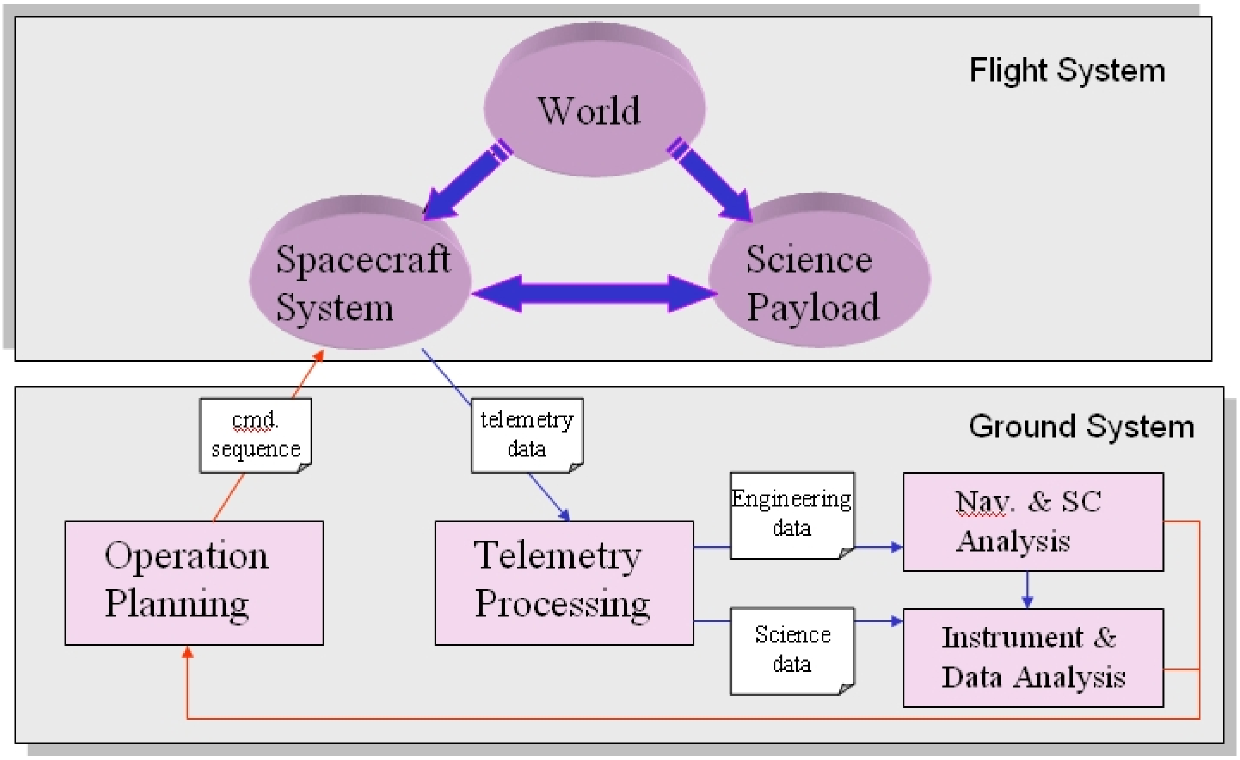

Figure 5. Flight System and Ground System Relationship during Operation Phase

A. Virtual Prototyping [2]

A "virtual prototype" is a special form of an engineering model that can be used in time-based operation simulation. The virtual world is composed of the stars, the Solar System, and the science targets. Each object in the virtual world observes its orbit dynamics and rotation period to accurately represent geometric and photometric properties at a specified time. The virtual spacecraft system is composed of navigation, attitude and articulation control, power, and telecom subsystems. The navigation system estimates the relative position and velocity of the spacecraft with respect to the world. The attitude and articulation control system points and tracks the spacecraft and gimbaled devices at specified directions. The power system charges batteries and distributes the power to the devices corresponding to the device states. The telecom system communicates with the Deep Space Network antennas on the Earth as well as specified relay stations. Finally, the virtual payload is composed of a wide range of instrument types including camera, spectrometer, and radar. Each instrument performs measurement process of the science target phenomena which includes target pointing, data product generation, resource usage profiling, and instrument state tracking.

All of the virtual prototype components described above can be automatically constructed employing three levels of class expansions, S class for information data structure, M class for situational analysis functions, and V class for operation command execution. For example, construction of a virtual prototype of a power subsystem involves S_power, M_power, and V_power classes where S_power class captures the solar panel and battery capacity, M_power provides a function that computes charge rate at a specified time, and V-power executes "turn on camera." The S classes are used to interface with the Excel subsystem templates directly or via XML scripts. The M classes are derived from the S classes and the V classes point to the M classes. The VMOF defines a pseudo command dictionary that lists all of the commands that the current V subsystem classes can perform. Figure 6 depicts the relationship between S, M, and V class libraries. The role of the P-class library in operation planning and the role of T-class library in telemetry generation are discussed below.

B. Operation Planning [7]

The VMOF supports three types of activities for operation planning, surface mapping, target observation, and downlink. For the surface mapping activity, the surface condition, mapping duration, and off-boresight pointing angle can be specified for controlling the data quality and volume. For the target observation, search area, surface condition, look angle, multi-frame mosaic option, and exposure duration can be specified. For the downlink operation, antenna pointing condition, data volume, and ground station can be specified for resource sharing and downlink opportunity assessment.

The desired observation specifications described above have a direct relationship to the mission parameters and the subsystem properties. The preliminary analysis is to verify the observation specifications against the subsystem properties and provide instantaneous reporting on any performance or resource conflicts. The conflicts may be resolved by changing the subsystem design or the science-return requirements. Comprehensive preliminary analysis requires a mapping between the observation conditions and the subsystem properties and interdependencies of the subsystem properties. In the surface mapping case, the surface condition is used to obtain the valid time range for the observation. The valid time range (combined with the instrument resolution) provides total data volume obtained during the surface mapping. The data volume and the orbit allocation for downlink determines required downlink rate, which in turn requires antenna properties. In the target observation case, a long exposure duration required due to either low instrument sensitivity or low target albedo may drive the attitude control system design if the instrument is body fixed in order to minimize the motion blurring. The target size, the search area, and the field-of-view of the instrument all affect the gimbal design in order to capture the target in the instrument’s frame as the spacecraft orbits.

The observation activity is composed as a set of subsystem events. Each subsystem event is described with an operation statement and a condition statement. The event operation is either a subsystem command or a macro command that can be decomposed as a set of subsystem commands. The list of supported subsystem commands is made available via a command dictionary. The event condition is defined as a logical combination of three types of conditions—target condition, time condition, and command condition. The target condition is used to express the necessary target state during the event operation. The supported target states include distance, apparent size, phase angle, and a number of others. The time condition is used to express the required time between events within a subsystem. The command condition is used to express interdependency and concurrency of the events among the multiple subsystems. In order to resolve the event conditions of each activity (target condition, time condition, and command condition) a set of condition analysis software modules is employed.

The S- and M-class libraries are utilized to formulate a P-class library for predicting the resource usage, resource availability, and resulting state of a subsystem. For example, P-telecom predicts power usage, downlink rate, and resulting antenna direction for a specific downlink command. The P-classes are employed to estimate the range of time when the event conditions can be satisfied. By combining the satisfied time ranges and the command modes, an operation sequence file is composed, and it is uplinked to the virtual prototype spacecraft system.

B. Time-based Spacecraft Operation Simulation [8]

The virtual spacecraft system is composed of the five subsystems, command and data handler, navigation, attitude and articulation control, power, and telecom. The command and data handler provides command sequence management, fault recovery, and telemetry record generation. The telemetry records are generated by sampling the subsystem states returned by each subsystem. Each subsystem shares a common state engine with three basic states, Ready, Command, and Busy indicating that a subsystem can receive a new command, a subsystem has received a new command, and a subsystem is in the middle of executing the previously received command. The virtual spacecraft simulates the operation state of a spacecraft by propagating each subsystem state as the spacecraft clock progresses the time in a regular interval. The subsystem state propagation is performed based on the operation rate and operation duration. The operation rate may be speed of a spacecraft, turn rate of a gimbal, input/output rate of a buffer, and so on. Various system noises can be also simulated by specifying error range and distribution.

The V-class library is used for operation sequence execution and telemetry generation. For example, V-AACS (Virtual Articulation and Attitude Control System) is responsible for “Point”, “Track”, and “Slew” commands for all of the articulated devices including antennas, solar arrays, and scan platforms. A T-class library defines the telemetry record structure for each subsystem to record the operation behavior of the virtual prototypes so that the operation behavior can be reconstructed on the ground for comprehensive monitoring of the interaction between the flight system and the world. The T-Navigation records time, spacecraft position, and spacecraft velocity relative to a reference system (e.g. the Sun). The T-AACS records attitude of the main body and a list of articulated devices For each device, device name, target name, intersect location and the gimbal information are recorded. The gimbal information includes the number of gimbal axes, the gimbal angles, and the pointing direction. The T-Payload records a list of instruments. For each instrument in operation, the T-Payload records instrument name, target name, intersect location, data file name, and the gimbal information. The T-Telecom records the antenna name, the file name that is being downlinked, and the downlink rate. The T-Power records a list of activated devices with their power usage status.

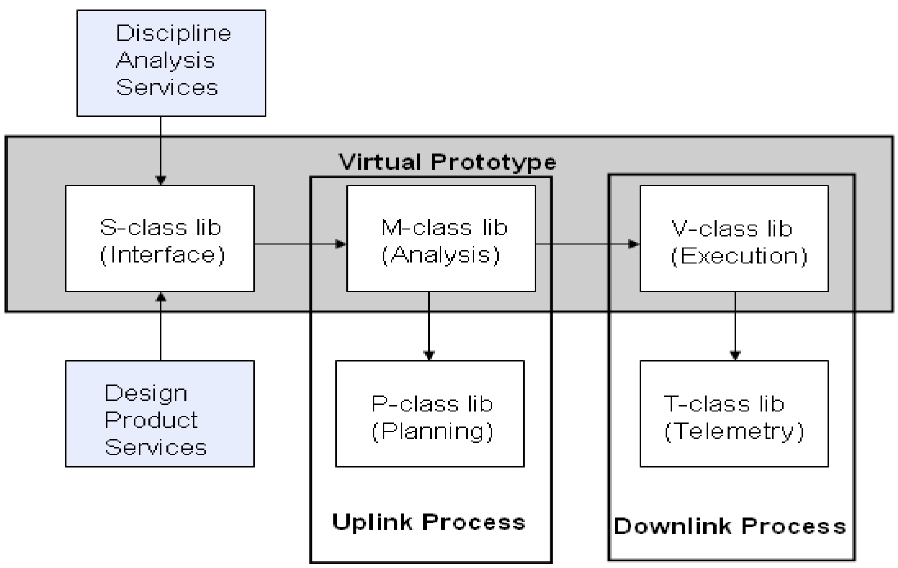

Figure 6. VMOF Software Infrastructure

IV. Distributed Visualization Service Framework

The Visualization Framework is composed of a Visualization server and a set of Visualization clients. “Micro-Helm” invented by Mission Simulation Group at JPL (shown in Figure 7), and it is employed for synchronously operating multiple visualization clients. The “Micro” indicates the PC-cluster architecture and the “Helm” indicates the comprehensive monitoring capability of the spacecraft operation. The Micro-Helm hosts a telemetry data server and a set of visualization clients that are organized along the subsystem architecture discussed earlier. Each subsystem visualization client employs one or more graphical windows to project the subsystem states relating them with the graphical models of the physical world and spacecraft structure. The graphical windows are managed by two servers, Overlay server and Animation server. The Overlay server manipulates two-dimensional (2D) images and plots while the Animation server provides three-dimensional (3D) solid-model rendering.

A. Real-time Telemetry Visualization [9]

The Visualization Framework can also serve real mission telemetry streams. Figure 7 is a snapshot of Micro-Helm during the Mars-Odyssey mission. The six screens display second-by-second spacecraft system states, telecom activities (upper left), spacecraft attitude with respect to the Sun and the Earth (upper middle), spacecraft boresight with respect to Mars (upper right), orbit trajectory and instrument views (lower left), spacecraft position and star tracker view (lower middle), and instrument orientations with respect to and their constraint conditions (lower right).

The subsystem Visualization clients support two modes, an initialization mode and a visualization mode. During the initialization mode, it establishes the socket connection with the telemetry data server and receives the subsystem design information as described in the S subsystem classes for the static properties of the spacecraft system. During the visualization mode, it receives the subsystem states from the Visualization server which translates the subsystem states received from the Telemetry server into graphical transforms that can be applied to the graphical models. Visualization of each subsystem state may employ 2D or 3D graphical representations of the world.

The navigation state visualization provides three types of trajectory displays for projecting the spacecraft position during three types of mission phases: cruise, orbit insertion, and in-orbit. The cruise-phase trajectory renders the geometric relationship of the spacecraft with respect to the Earth and the target. The orbit-insertion-phase trajectory renders the geometric relationship between the spacecraft and the target. The in-orbit-phase trajectory renders the orbit track on the target surface. The attitude and articulation state visualization provides the spacecraft orientation and the pointing direction of the articulated devices. The coordinate axes, the Sun direction, the Earth direction, and the spacecraft-nadir direction are indicated to help in validating the orientation of the spacecraft and the articulated devices. The telecom state visualization provides a display of an active Deep Space Network (DSN) complex during the downlink operation with the background of the three DSN complexes and their coverage ranges. The power state visualization provides a continuously running plot of the available power, the power consumption profile of each device, and the state of the device. The instrument state visualization provides swath footprints on the target surface for push-broom operation during surface mapping and field-of-view footprints for the target observation. In addition to the footprint overlay, a surface map is updated for the coverage status.

B. Synchronized Multi-Channel Broadcasting

Multiple windows support comprehensive viewing of mission states, and this viewing, can be easily shared among multiple design teams. In order to support distributed design teams, the Visualization Framework is being extended to serve multiple Micro-Helms simultaneously. Synchronization among multiple Micro-Helms requires controlled digital streaming which involves multiple buffering and a global clock. During Cassini Saturn Orbit Insertion (SOI), six views of the SOI events were organized into one display unit with six channels. The display unit was made available to distributed teams around the country to receive the predicted mission states that were synchronized with the Earth-Receiving-Time of Cassini over an 8-hour period. It was the first time a space mission shared its encounter events with the general public in real-time. The distributed mission state visualization capability combined with real-time mission simulation capability enables collaborative design among multi-disciplinary teams remotely. Figure 9 illustrates the spectator view channel of the display unit developed for Cassini Mission Simulation Visualization.

Figure 7. Mars-Odyssey Micro-Helm

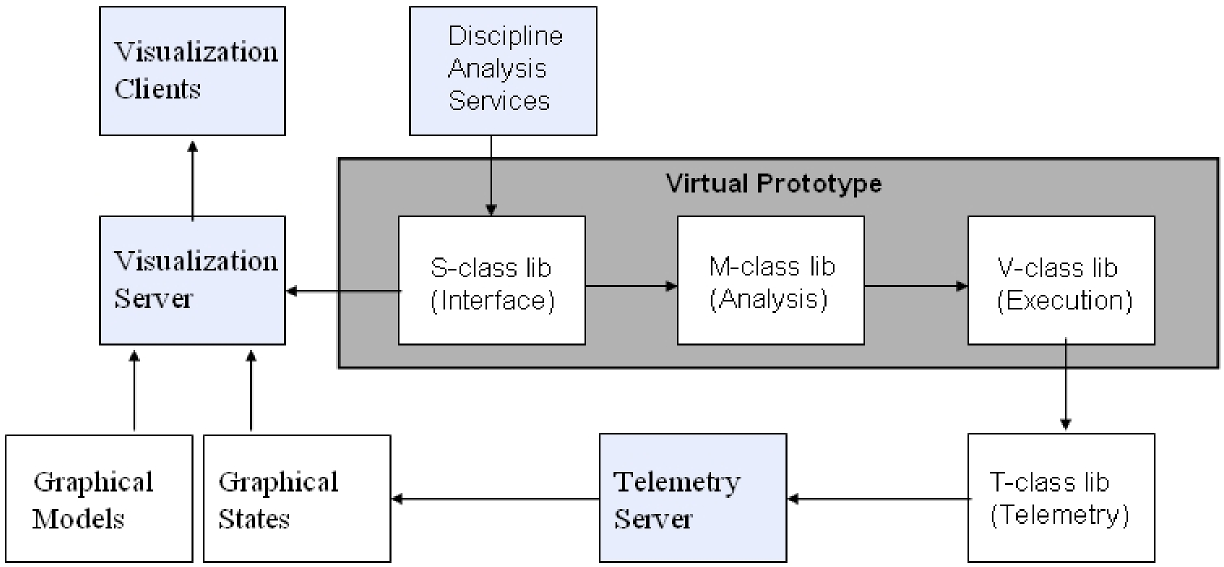

Figure 8. Relationship of Visualization Server, Telemetry Server, and Virtual Prototype

Figure 9. Mission Simulation Visualization Network for Cassini-SOI

V. Conclusion

The VMOF enables heterogeneous model integration, rapid virtual prototyping, automated operation planning, high fidelity operation simulation, and comprehensive operation visualization. By introducing the operation-centric view of a mission during the formulation phase, concurrent engineering between system design team and the operation design team can be achieved, mitigating late discoveries that could compromise science-return or jeopardize mission success. By enabling distributed heterogeneous model-integration mechanisms, a system-level view can be developed in collaboration with the subsystem discipline models without compromising the subsystem-level optimality. By providing real-time simulation visualization with respect to multiple discipline teams, all teams can share an accurate view of the mission objectives and quickly grasp the design impacts on each other.

The VMOF team is extending the architecture to become a multi-discipline collaborative engineering framework in support of the Model-Based Engineering Design (MBED) initiative at JPL. The initiative is to enable advanced systems engineering practice through a series of integrated, increasingly detailed models that provide continuity from architectural concept through detailed design. It will extend current capabilities for rapid conceptual design, allowing thorough exploration of design trade-spaces and selection of an optimal design point with associated cost and rationale; and it will provide seamless connection to subsystem models and detailed design tool suites. The VMOF team is also collaborating with Ames Research Center on “End-to-End In-Situ Mission Operation Simulation Framework” that simulates the mission operators’ workflow and decision making process.

Acknowledgments

This research was carried out at Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The research was partly supported by the Computational Technologies Program of NASA's Earth-Sun System Technology Office and JPL institutional Research and Technology Development Program. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] Lee, M. Weidner, R. J., "Science-Return Modeling and Simulation," *IEEE Aerospace Conference*, Big Sky, MT, Vol.8, pp. 3719–3730, 2003.

[2] Lee, M. Weidner, R. J., "Design-Based Mission Operation," *IEEE Aerospace Conference*, Big Sky, MT, Vol.7, pp. 3429–3442, 2001.

[3] Wall, S., "The Mission System Design Center: A Pilot of Formulation-Phase Concurrent Engineering in Aerospace Design," *Tenth Annual International Symposium of the International Council on Systems Engineering*, Reno, CA, 2001.

[4] Weidner, R. J., "A Component-Based Implementation of Agents and Brokers for Design Coordination," *IEEE Aerospace Conference*, Big Sky, MT, Vol. 6, pp. 2889–2895, 2001.

[5] Bigus, J. P., , *Constructing Intelligent Agents with Java*, New York: John Wiley & Sons, Inc., 1998.

[6] Pallmann, D., *Programming Bots, Spiders, and Intelligent Agents in Microsoft Visual C++*, Redmond, Washington: Microsoft Press, 1999.

[7] Wang, S., "SCENAREO- Science and Engineering Activity Reasoning and Optimization," *IEEE Aerospace Conference*, Big Sky, MT, Vol. 6, pp. 3834-3845, 2004.

[8] Lu, W., "TSOS – Time-Based Spacecraft Operation Simulator," *IEEE Aerospace Conference*, Big Sky, MT, Vol. 6, pp. 3827-3833, 2004.

[9] Lee, M., Weidner, R. J., "Science Observation and Mission Lifecycle," *SpaceOps-2002*, Houston, TX, 2002