AN INTEGRATED APPROACH TO PAYLOAD SYSTEM SIMULATION

Meemong Lee
Raymond L. Swartz, Jr.
Ansel Teng
Richard J. Weidner

MS168-522 Jet Propulsion laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, Ca. 91109
meemong@elroy.jpl.nasa.gov

Sept. 15, 1996

Abstract

This paper describes a payload system simulation implemented at JPL, as part of a comprehensive mission simulation facility. The payload system simulation is responsible for providing the virtual presence of a payload system by simulating its behavior with respect to flight software, spacecraft system architecture, and mission data product generation. The purpose of the virtual presence is to enable concurrent mission design, integration, and validation. The transition from the virtual payload system to a real payload system is transparent to the rest of the system.

The flight software function includes communication with other process modules, instrument control, and data management. The spacecraft system architecture function includes resource consumption profile, operational constraints, and bus interface. The mission data product generation function includes target phenomena model integration, radiometrically accurate sub-pixel-level ray tracing, instrument response characteristics simulation, and mission environment impact simulation.

The payload system simulation software consists of: a *camera subsystem*, which performs flight software functions and simulates the response behavior of an instrument for mission data product generation, a *virtual world*, which synthesizes radiometric and geometric relationship between the sun, a target body, and the payload system during a specified mission period; and a *mission visualization toolset*, which provides comprehensive monitoring of the various aspects of the dynamic relationship among the instrument, the spacecraft, the target, and the mission environment.

1 Introduction

In order to significantly reduce the mission life cycle from proposal to launch, the mission phases of design, development, integration, and test need to be optimally coordinated. An essential element of the coordination is an integration and test environment where a design concept can be easily tested for its over all impact, a prototype system can be operated in a realistic mission setting, and flight/ground software can be integrated and benchmark ed.

The JPL Flight System Testbed (1''S'1') [1] has been working toward providing such an environment. The FST is composed of five systems: sequence manager, articulation] and attitude control system (AA(X), telemetry system, power system, and payload system. Each of the five systems provides both a flight component and a simulation component. 'J']](! flight components of the five systems are integrated as a flight software unit, and the simulation components are integrated as a support equipment unit. The integration is implemented via Tramel inter-process communication protocols [2].

This paper discusses an integrated approach to simulating a payload system. This approach has been implemented at, the Flight System Testbed. The Payload system provides camera flight software module, camera support equipment module, and a visualization toolset. The camera support equipment module requires simulation of the camera system response as well as the environment the camera will observe. Thus, the payload system is described in three parts, the *Virtual World*, the *Camera Subsystem*, and the *Mission Visualization Toolset*.

The *Virtual World* is a physics-based model simulation which describes the world which a payload system will encounter during a mission. It provides the basis for geometrically and radiometrically accurate mission data product generation. The accuracy of the synthesized mission data products is important to validate on-board and ground data processing and analysis operations. The accuracy in this context implies the computational fidelity of the light path tracing and photon count integration based on the phenomena model.

The *Camera Subsystem* is software which consists of a flight module and a support module. The flight module interfaces (commanding and data retrieval) to the instrument's electronics and to other subsystem modules of the spacecraft flight software. The support module simulates camera system behavior and the mission data product generation process interacting with the virtual world.

The *Mission Visualization Toolset* comprises graphical utilities which are used for displaying various aspects of mission operation relevant to the payload system sequence. The display allows validation of payload sequence designs (with respect to spacecraft dynamics, attitude control, and data acquisition), as well as communication of the mission concept to a wider community.

2 Virtual World

The Virtual World is composed of two parts: a phenomena model database, and an ideal world view generator. It is implemented to provide geometrically and radiometrically accurate mission data products so that they can be used by developers of on-board algorithms, as well as ground data systems. The geometric accuracy is defined as the accuracy of the apparent location, shape, and size of the observed bodies, and the radiometric accuracy represents the accuracy of the contrast and absolute intensity level of the pixels in the simulated data products.

2.1 Phenomena Model Database

The intent of the Phenomena Model Database is to establish a link to the state-of-the-art scientific knowledge database so that the body of knowledge obtained from previous missions can be optimally utilized for future missions. The body of scientific knowledge includes measurements, models, hypotheses, etc.

For a given mission, the world is represented by the science target bodies, navigational target bodies, stars, and the cosmic environment. The phenomenon of a planetary body is represented in the database to meet the observational level of detail; thus the level of complexity varies depending on the payload system and mission scenario. For example, a body may be described simply with a set of orbit elements, average albedo, and tri-axial ellipsoid body shape parameters. Or it can be described in detail with a surface texture map, digital elevation map, material composition map, etc.

The mission and payload system dependencies are represented by employing a hierarchical database structure. The phenomena model database structure consists of an orbit dynamics model, photometric model, geological model, and atmospheric dynamics model.

- *orbit dynamics model* The orbit dynamics of a body is represented in SPICE data format [3], invented at JPL for archiving mission events. The orbit dynamics file is generated by propagating the orbital elements, semi-major axis, eccentricity, inclination, longitude of the ascending node, argument of periapsis, true anomaly at a specified time, and gravitational force. The Object-oriented SPICE library (OO SPICE) [4] is employed to propagate the body state at an arbitrary time and to compute a relative state from an arbitrary reference coordinate system.

- *photometric model* The photometric model of a body is represented with an albedo map and photometric function (Lambert, Hapke, etc.). For bodies with dynamic photometric characteristics (such as a coma), the photometric model is described as a function of time and distance.

- *geological model* The geological model is for the surface characteristics of a solid body, and it is described with a texture map, digital elevation

map, and material map. The texture map is for the surface albedo variation in the visible wavelength range; the digital elevation map is for the surface height variation from the mean elevation; and the material map is for surface material composition and distribution.

- *atmospheric dynamics model* The photometric characteristics of a gaseous body or the gaseous part of a body (i.e., al) atmosphere 01' a cometary coma), are described with a body specific material composition and a dynamics model provided by the science community. The model incorporates particle size information and phase functions of materials, as well as heterogeneous gaseous structure. A generic mechanism for integrating atmospheric structure and dynamics will be researched.

2.2 ideal World View Generator

After establishing the phenomena database of a specific mission world, the observation sequence of the world is simulated in two parts: observed view with an ideal instrument, and distortion of the acquired view by a real instrument. The ideal instrument simulation implies performing the scene generation following the camera viewing geometry without introducing the camera distortion characteristics.

The ideal world view generation involves ray tracing of the reflected solar energy from the planetary bodies and stars. The ray tracing may be performed using multiple rays for each pixel. The number of rays per pixel is determined based on the analysis to be applied to the mission data products. Algorithms analyzing the target locations to sub-pixel accuracy for optical navigation may require up to 1 00 rays per pixel.

The ray tracing requires a comprehensive viewing geometry between the spacecraft, camera, target bodies, sun, and stars. In order to simulate the spacecraft motion during the exposure, it is necessary to perform the ray tracing often enough to capture the relative motion between the spacecraft and the target body. The results of the ray tracing are integrated over the exposure duration to compute the total radiation arriving at the instrument. The integrated image is referred to as the *ideal scene* (implying the absence of instrument distortion). It represents the world the instrument sees. A *real scene* is then simulated by applying the sensitivity of the detector and adding the distortions from the instrument's characteristics with respect to optics, detectors, cosmic rays, readout logic, and electronics.

3 Camera Subsystem

The Camera Subsystem is responsible for command and data-handling of a payload system during observation. It is designed to support a virtual instrument phase and a real instrument phase, with minimal change between the two phases. The virtual instrument refers to the instrument behavior simulation, while the real instrument refers to the real electronics and detectors.

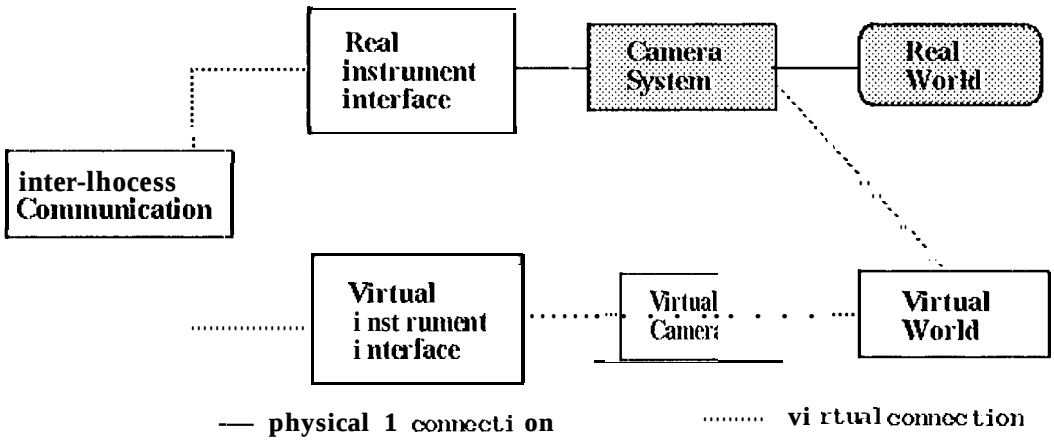

Figure 1: Payload System Software Architecture

In order to easily transition from the virtual instrument phase to the real instrument phase, the camera flight software is implemented in two submodules, instrument-specific and instrument-generic. The instrument-specific submodule is implemented for a real instrument to control the instrument's electronics system; and for a virtual instrument, to send data acquisition commands to the scene generator.

The generic submodule performs inter-process communication and data handling. The virtual instrument interface submodule is responsible for simulating spacecraft's resource usage during operation (in terms of power, bus, and storage). Figure 1 describes the virtual instrument phase and the real instrument phase. The dotted line between the Camera System box and the Virtual World box indicates that the Virtual World simulation may be placed in front of a real camera system to support the integration and test, process in the real instrument phase.

3.1 Instrument Response Simulation

The response/distortion characteristics of an instrument are represented with a set of parameters describing system aspects as well as component aspects.

- System Parameters — Wavelength Range, Point-Spread Function
- Optics Parameters — Aperture size, F-number, Focal Length
- Detectors — Array sire, Pixel sire, Full well, Quantum efficiency, Dark current
- Electronics — A/D converter, Read noise

The parameter set listed above is the basic descriptor set which applies to all imaging instruments. Instrument-specific features are described in an extended parameter set. The extended parameter set includes descriptors for auxiliary

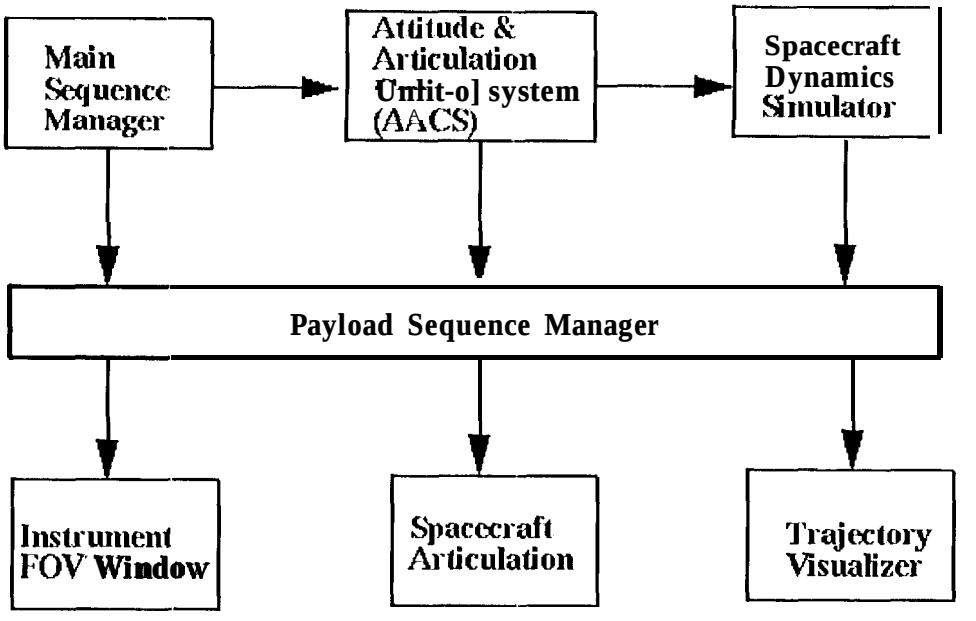

Figure 2: 1 'ayload Sequence Manager and the Visualization Toolset

devices such as a filter w heel, mirror, or shutter, and for special features such as the blooming featu re of' CC]) detectors, special purpose buffers, etc.

4 Mission Visualization Toolset

The toolset consists of a trajectory visualizer, spacecraft articulation, and an instrument FOV window n ıanager. The three visualizers are synchronized via a payload sequence manager. The payload sequence manager receives requests from the main sequence manager for data acquisition and from the Articulation and Attitude Control System (AACS) for articulation of' solar panels and other movable parts. It subscribes the achieved spacecraft's Static (position and attitude) from the spacecraft dynamics simulator. Figure 2 illustrates the inter-process message publish/subscribe relationship.

The toolset is built using the C-+ language and OpenInventor 3D graphics library [5]. The mission-indep e ndency of the toolset is achieved by reading al] mission-specific information from mission description files. All views of a given body are based 011 the same database, but the size of the body in each view can be scaled independently to achieve the best visual effect.

4.1 Trajectory Visualizer

The trajectory visualizer was implemented to overview the mission scenario with respect to the global relationship between the Sun, target bodies, and spacecraft, as well as the local relationship during a dose approach to a target body. The visualization requires manipulation of 4-dimensional space and time information. A snapshot of the trajectory visualization for the Micro

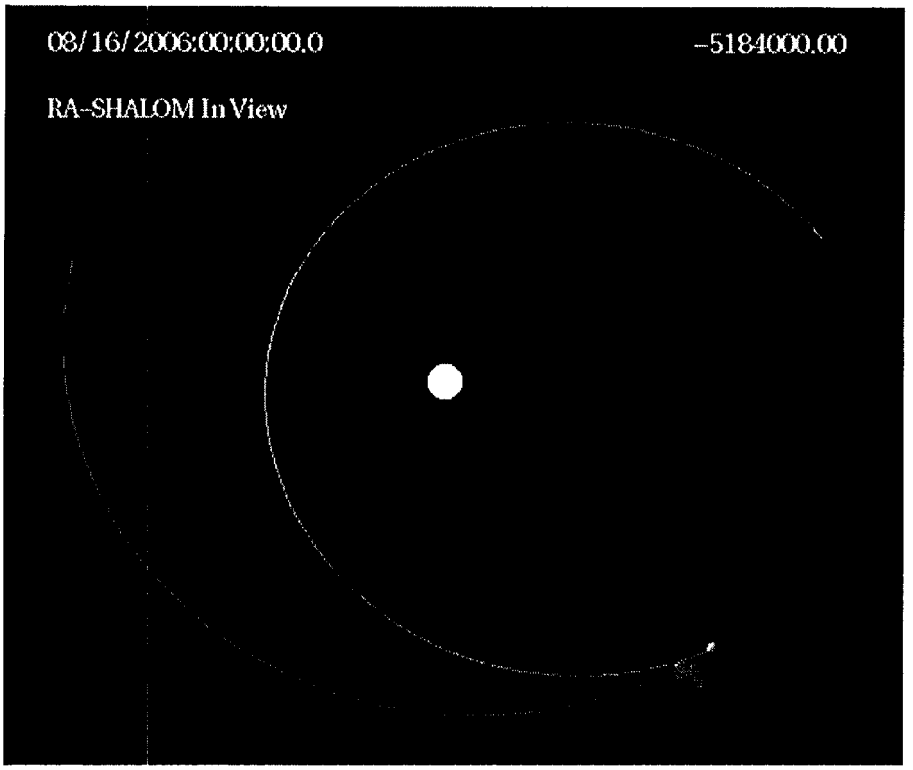

Figure 3: Snapshot of Micro Spacecraft Trajectory Visualization

Spacecraft is illustrated in Figure 3.

The trajectory visualizer provides interactive 31) coordinate transform, dynamic reference coordinate change, and dynamic state propagation time step change, and payload sequence event labeling. The tool accepts mission specific information via a set of files/messages organized as shown below.

- Target body

  orbit dynamics : the target body orbit is precomputed in a SPICE-format file as described in the orbit dynamics model in the phenomena model database.

  - body shape : the body shape is described in two steps; the first step as a tri-axial ellipsoid, and the second step as a digital elevation map in an OpenInventor file format. The digital elevation map represents the local area deviation from the triaxial ellipsoid in a polygonal mesh.
  - body texture : the body surface albedo of a given wavelength range is described in a cylindrical map. The albedo file is used when the observation distance is close enough to require surface feature visualization.

- Spacecraft

  trajectory : the spacecraft trajectory may be precomputed as a SPICE-format file similar to the orbit dynamics file of the target

body. Or the spacecraft state (position and attitude) may be received from a spacecraft dynamics simulation module via a Tramel message.

- SC model : the default spacecraft model is described as a simple globe with specification of color and size. A detailed structure may be introduced by providing an OpenInventor file.

- 1 Events

  visualization control : the dynamic setting of the reference target, time step, and the special mission events arc provided via an event file.

  - payload sequence : the data acquisition scenario during an encounter period is prearranged in a payload sequence file indicating required attitude control and payload system control.

4.2 Spacecraft Articulation

The main objective of the spacecraft articulation tool is to visualize the spacecraft geometry during the payload sequence, including the spacecraft's stability, orientation of the payload system relative to the Sun, and obscuration of the field of view. Ultimately, this tool will incorporate structural and thermal effects on the payload sequence to monitor the instrument state in various operational environments.

Currently, a close-up view of the spacecraft is provided in a separate window, displaying attitude change, articulation of the solar panels, and the behavior of various moving parts. The articulation of a part is expressed as a three dimensional position and rotation at a specified time. The spacecraft articulation window is synchronized with the trajectory visualization window and the instrument's FOV window so that a comprehensive monitoring of spacecraft behavior can be achieved.

4.3 Instrument FOV View

In older to appreciate the instrument's viewing geometry relative to the target motion surrounding the exposure command, a wide-angle camera which contains the actual instrument's FOV is synthesized. The field-of-view of each individual sensor is marked on the viewer as a proportionally scaled box. When a sensor is in the exposure mode, the corresponding field-of-view box indicates the exposure action by blinking.

This view provides effective visualization of minute changes in spacecraft state during the payload sequence execution that are not visible from the spacecraft articulation window. The deviation from the desired state due to the errors incurred by the navigation and attitude control system are apparent in this window.

The wide-al]glc camera view is particularly useful when an instrument has multiple FOVs, as in the case of MICAS (Miniature Imaging Camera and Spec-

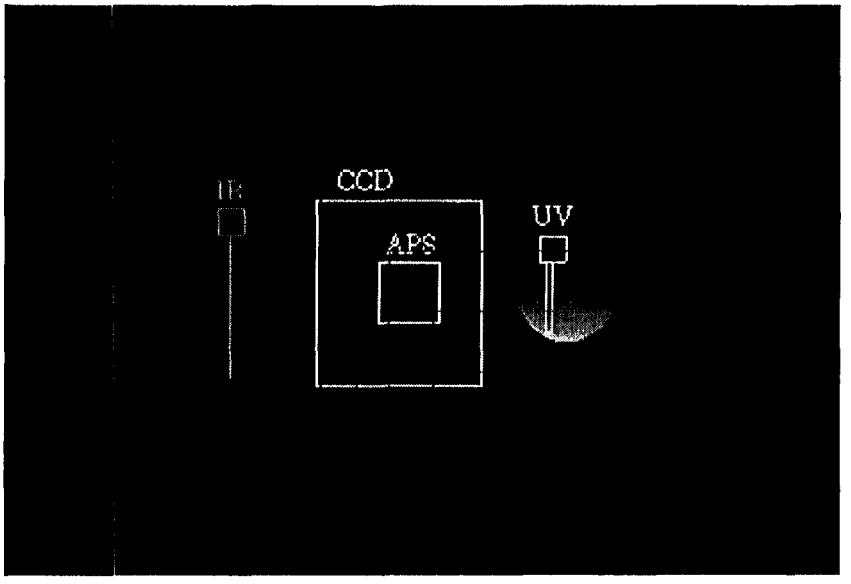

Figure 4: Geometric Alignment of Fields-of-View of Four Sensors of MICAS

trometer) [6]. MICAS has four detectors sharing the same optics: CCD (Charge-Coupled Device), APS (Active Pixel Sensor), UV (Ultraviolet Spectrometer), and IR (Infrared Spectrometer). The field-of-view of each sensor maps the sky uniquely, requiring sensor-specific AACS control to point to the target. Figure 4 illustrates the alignment of the four sensors and the target body pointing.

Figure 5 is an example data product acquired by a simulated CCD sensor of MICAS during the virtual asteroid encounter.

5 Future Research

The payload system simulation described in this paper has been utilized in support of several projects including Stardust, Pluto Fast-Flyby, Micro Spacecraft, and New Millennium/DS-1. The role of the payload system has been high-fidelity mission data product generation for on-board autonomy algorithm validation, payload sequence design validation, and mission concept visualization.

In the future, the payload system simulation will be extended to support instrument design and development efforts more directly. The extension will apply to all three components of the payload system discussed above by pursuing the following three areas:

- Target body Synthesis Build a comprehensive and multi-resolution target body phenomena 1110(1)1 database for an efficient integration of the entire body of state-of-the-art scientific knowledge. Implement a real-time model simulation on a massively parallel architecture to support an end-to-end mission simulation.

- Interface design Infrastructure Design/development of a universal elec-

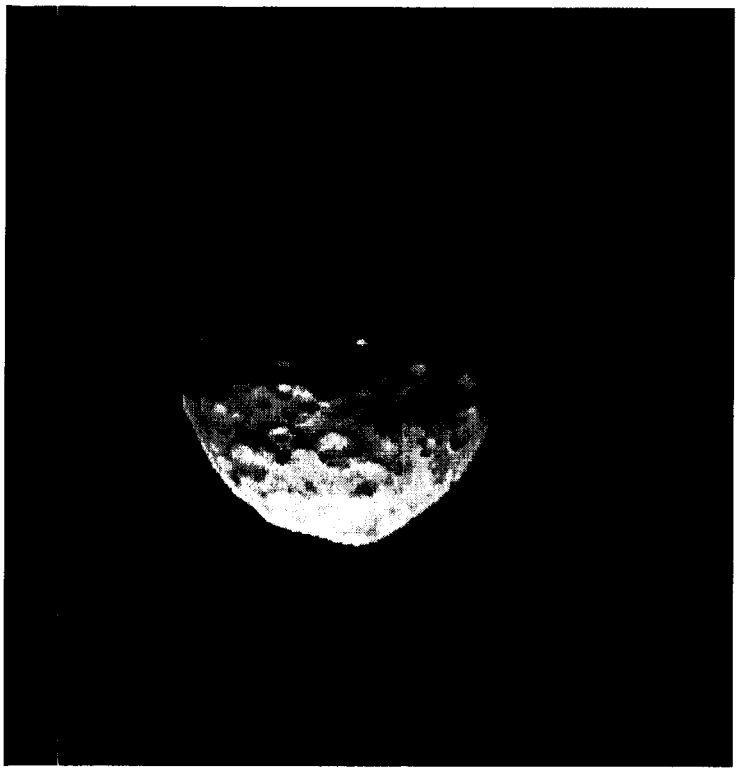

Figure 5: Target Asteroid Image Acquired by the CCD Sensor of MICAS

tronics framework for integrating various detectors so that the detectors' performance can be tested and calibrated readily. The electronics framework will include instrument-resident processor architecture for *smart* instrument design.

- Payload sequence design architecture Adaptation of advanced 3D graphics and virtual reality technology for the design of instrument structure and articulation and for the design of payload system operational scenarios.

Acknowledgement

This task was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The task was partly funded by the *Information Systems Office* within the Office of Space Science, NASA.

References

[1] JPL Flight System Testbed Prototype Team, JPL Flight System Testbed Prototype System, *http://jpl/fst/proto.html.*

[2] Scott Burleigh, Task Remote Asynchronous Message Exchange Layer, *http://fst/Products/Tramel/Manual.*

[3] Charles Acton, Jr., Ancillary Data Services of NASA's Navigation and Ancillary Information Facility, *Planetary and Space Sciences, Vol. 44, No. 1 PP 65-70, 1996*

[4] Richard Weidner, Object-Oriented SPICE Library, *oospice under http://spartacus/ richard*

[5] OpenInventor Architecture Group, OpenInventor C++ Reference Manual, Release 2, *Addison-Wesley Publishing Company.*

[6] Amit Sen, Interface Control Document & Product Functional Specification Document, Sept, '96, *NMP DS1 6446-464, JPL D-13543*