

1315/5

Modular Software for Spacecraft Navigation Using the Global Positioning System (GPS)*

S. H. Truong, K. R. Hartman, D. A. Weidow, and D. L. Berry
National Aeronautics and Space Administration
Goddard Space Flight Center (GSFC)
Greenbelt, Maryland, USA 20771

D. H. Oza, A. C. Long, E. Joyce, and W. L. Steger
Computer Sciences Corporation (CSC)
Lanham-Seabrook, Maryland, USA 20706

Abstract

The Goddard Space Flight Center (GSFC) Flight Dynamics and Mission Operations Divisions have jointly investigated the feasibility of engineering modular Global Positioning System (GPS) navigation software to support both real-time flight and ground postprocessing configurations. The goals of this effort are to define standard GPS data interfaces and to engineer standard, reusable navigation software components that can be used to build a broad range of GPS navigation support applications. The paper discusses the GPS Modular Software (GMOD) system and operations concepts, major requirements, candidate software architecture, feasibility assessment, and recommended software interface standards. In addition, ongoing efforts to broaden the scope of the initial study and to develop modular software to support autonomous navigation using GPS are addressed.

1.0 Introduction

The Department of Defense (DOD) Global Positioning System (GPS) is becoming a more attractive navigation option for National Aeronautics and Space Administration (NASA) spacecraft due to the recent declaration that the GPS is fully operational. The NASA Mission Operations and Control Architecture (MOCA) Program (Reference 1) has shown that GPS has the potential to provide significant economies for NASA spacecraft navigation.

The GPS civilian Standard Positioning Service (SPS) can deliver spacecraft navigation accuracies in the 20- to 100-meter (1σ) range in a real-time flight environment, improving to the submeter level in a postprocessing ground environment (References 2 and 3). The DOD intentionally limits the real-time navigation accuracy achievable using GPS SPS by applying Selective Availability (SA) corruption to the GPS signals. The GPS military Precise Positioning Service (PPS) can deliver spacecraft navigation accuracies in the 5- to 15-meter (1σ) range in a flight environment, but PPS access and use are restricted and subject to security classification requirements (Reference 2).

Currently, major drawbacks to the use of GPS on NASA spacecraft include the lack of standardization in the products available from commercial GPS receivers and the lack of reusable navigation flight software and ground support software. To promote rapid, cost-effective deployment of GPS technology, NASA's Goddard Space Flight Center (GSFC) Flight Dynamics Division (FDD) and Mission Operations Division (MOD) were jointly funded by NASA Headquarters/Code OI to perform a quick-reaction study to define the standard GPS data interfaces and to investigate the feasibility of engineering modular software components to support both flight and ground GPS navigation applications.

* This work was supported by the National Aeronautics and Space Administration (NASA)/Goddard Space Flight Center (GSFC), Greenbelt, Maryland, under Contract NAS 5-31500.

The core GPS Modular Software (GMOD) team consisted of representatives from the GSFC MOD flight software development, FDD ground software development, and FDD operations and analysis disciplines. Draft materials were distributed to NASA Headquarters/Code OI, Johnson Space Center (JSC), and Jet Propulsion Laboratory (JPL) representatives for review.

Due to the short timeframe of the initial study (i.e., 3 months), the scope of the study was limited to single-spacecraft navigation using a commercial GPS SPS receiver. All major study objectives were accomplished:

- A complete end-to-end candidate system was defined to provide single spacecraft autonomous navigation using a single frequency GPS SPS receiver.
- GPS data interface standards were recommended.
- Operational configurations were recommended that logically partition GPS navigation functions between flight and ground segments to support several levels of spacecraft autonomy.
- An open system architecture design was developed that provides the flexibility to host standard navigation software components onboard or on the ground to provide a range of mission-selectable flight/ground functional partitions.

Based on this investigation, the GMOD team did not identify any inherent obstacles to the development of an open system architecture that partitions the GPS navigation software between flight and ground. Follow-on activities are underway to extend the scope of this initial study to address use of the GPS PPS, multiple spacecraft navigation applications, and high-accuracy postprocessing applications and to demonstrate the feasibility of the GMOD concepts.

Reference 4 provides a detailed discussion of the results from the initial GMOD feasibility study. This paper summarizes these accomplishments. Section 2.0 discusses the system and operations concepts, Section 3.0 addresses the software architecture design accomplishments, and Section 4.0 lists follow-on activities resulting from this study.

2.0 System and Operations Concepts

The following GMOD system goals were defined for this initial study:

- The GPS Modular software will support the following navigation functions: geometric position computation, state (i.e., spacecraft position and velocity and GPS receiver time bias) estimation, state prediction, orbit maintenance, and navigation performance monitoring and calibration for a single user spacecraft.
- The modular software will support real-time and near-real-time autonomous navigation operations to provide spacecraft navigation accuracies in the 20-meter (1σ) range using a single-frequency commercial GPS SPS receiver, with SA at typical levels.
- The modular software will support several levels of autonomy ranging from ground processing of downlinked GPS receiver measurements to autonomous onboard state estimation and orbit adjustment.
- GPS modular software will require standardization of its external interfaces, such as GPS measurement interface, command interface, and telemetry interface. GPS data interface standards will be defined, and custom interface software will be needed to convert the output of the selected GPS receiver to the standard data interface format.

This section discusses the system and operations concepts for modular software that satisfies these goals.

2.1 Major Functional and Operational Capabilities

The following major GPS navigation support capabilities were identified based on the GSFC FDD's Tracking and Data Relay Satellite System (TDRSS) Onboard Navigation System (TONS) (References 5 and 6), GPS Enhanced Orbit Determination Experiment (GEODE) (References 7 and 8), TONS Ground Support System (TGSS) (Reference 9), and the institutional navigation and orbit maintenance software in use in GSFC's Flight Dynamics Facility (FDF):

- **Geometric Positioning**—Computing the spacecraft position and receiver's time bias by iteratively solving four simultaneous pseudorange measurement equations and computing the spacecraft velocity and receiver's time bias rate by solving four simultaneous pseudorange rate (derived from the frequency shift of the carrier signal) measurement equations. This purely geometric approach provides instantaneous "point" solutions at the measurement times. This capability is available within most commercial GPS receivers.
- **Spacecraft State Estimation**—Computing corrected estimates for the spacecraft position, velocity, receiver time bias, receiver time bias rate, and atmospheric drag coefficient by filtering pseudorange and/or Doppler measurements or, alternatively, geometric positioning states. This approach models the spacecraft dynamics and therefore does not require continuous tracking of four or more GPS space vehicles (SVs) simultaneously.
- **Spacecraft State Propagation**—Predicting the spacecraft position, velocity, and time bias ahead in time using a high-accuracy model of the spacecraft dynamics.
- **Orbit Adjustment**—Identifying when an orbit maneuver is required to maintain the orbit within mission-specific constraints (typically arising from frozen orbit, Sun-synchronous orbit, or ground-track repeatability requirements for low Earth orbit (LEO) missions); determining the time, location, magnitude, and direction of the maneuver; and computing the associated thruster on/off times.
- **Navigation Performance Monitoring and Calibration**—Verifying the quality of the GPS measurements and the accuracy of the spacecraft state estimation and orbit adjustment computations and calibrating models to improve navigation performance.

2.2 Navigation Configurations

Table 1 lists a set of GPS navigation configurations, developed based on Reference 1, that provide increasing levels of autonomous onboard functions and require decreasing levels of ground support functions. The total set of navigation support functions remains essentially the same, regardless of the flight or ground computing environment. The GMOD team developed detailed descriptions for the first four navigation configurations, addressing the major functions and associated external and internal data interfaces. This information is presented in detail in Section 2.2 of Reference 4. The last three configurations, which involve the use of differential GPS measurements from multiple spacecraft or possibly multiple ground stations, are beyond the scope of the initial GMOD study but are being addressed in a follow-on study.

Figure 1 illustrates the functional partitioning developed for the autonomous navigation configuration, which provides accurate real-time position and velocity computation onboard using a real-time extended Kalman filter estimation approach with a high-fidelity spacecraft dynamic model. This autonomous navigation configuration reduces the need for definitive ground state estimation to a backup function except for periodic validation and calibration of onboard performance. A major advantage of this configuration is that accurate position, velocity, and time estimates are available for autonomous precision attitude control and direct downlink with the science data, eliminating the need to uplink predicted ephemeris information to the spacecraft and to perform postfacto state estimation on the ground. In addition, real-time position and velocity estimates are available on the spacecraft to support reinitialization of the GPS receiver if required.

Table 1. Summary of GPS Navigation Configurations

Navigation Configuration	Onboard Functions	Ground Functions
Repeater Configuration	GPS signal digitization, storage, and forwarding*	GPS measurement/broadcast message extraction State estimation and prediction Navigation performance monitoring and calibration Orbit adjustment
Geometric Positioning Configuration	GPS measurement/broadcast message extraction* Geometric position computation**	State estimation and prediction Navigation performance monitoring and calibration Orbit adjustment
Autonomous Navigation	GPS measurement/broadcast message extraction* Geometric position computation** State estimation and prediction Navigation performance monitoring	Navigation performance calibration Orbit adjustment
Autonomous Orbit Maintenance	GPS measurement/broadcast message extraction* Geometric position computation** State estimation and prediction Navigation performance monitoring Orbit adjustment	Navigation performance calibration
Postprocessed Differential Navigation	GPS measurement/broadcast message extraction* Geometric position computation (optional)**	State estimation using differential pseudorange and phase measurements Navigation performance monitoring and calibration Orbit adjustment
Real-Time User Spacecraft Constellation Navigation	GPS measurement/broadcast message extraction* State estimation using differential ranges from neighboring spacecraft Navigation performance monitoring	Navigation performance calibration Orbit adjustment
Formation Flying/Rendezvous	GPS measurement/broadcast message extraction* Compute relative positions between two spacecraft using differential ranges Navigation performance monitoring Orbit adjustment to maintain range/close distance	Navigation performance calibration

Notes: * Standard receiver function
 **Provided by most commercial receivers

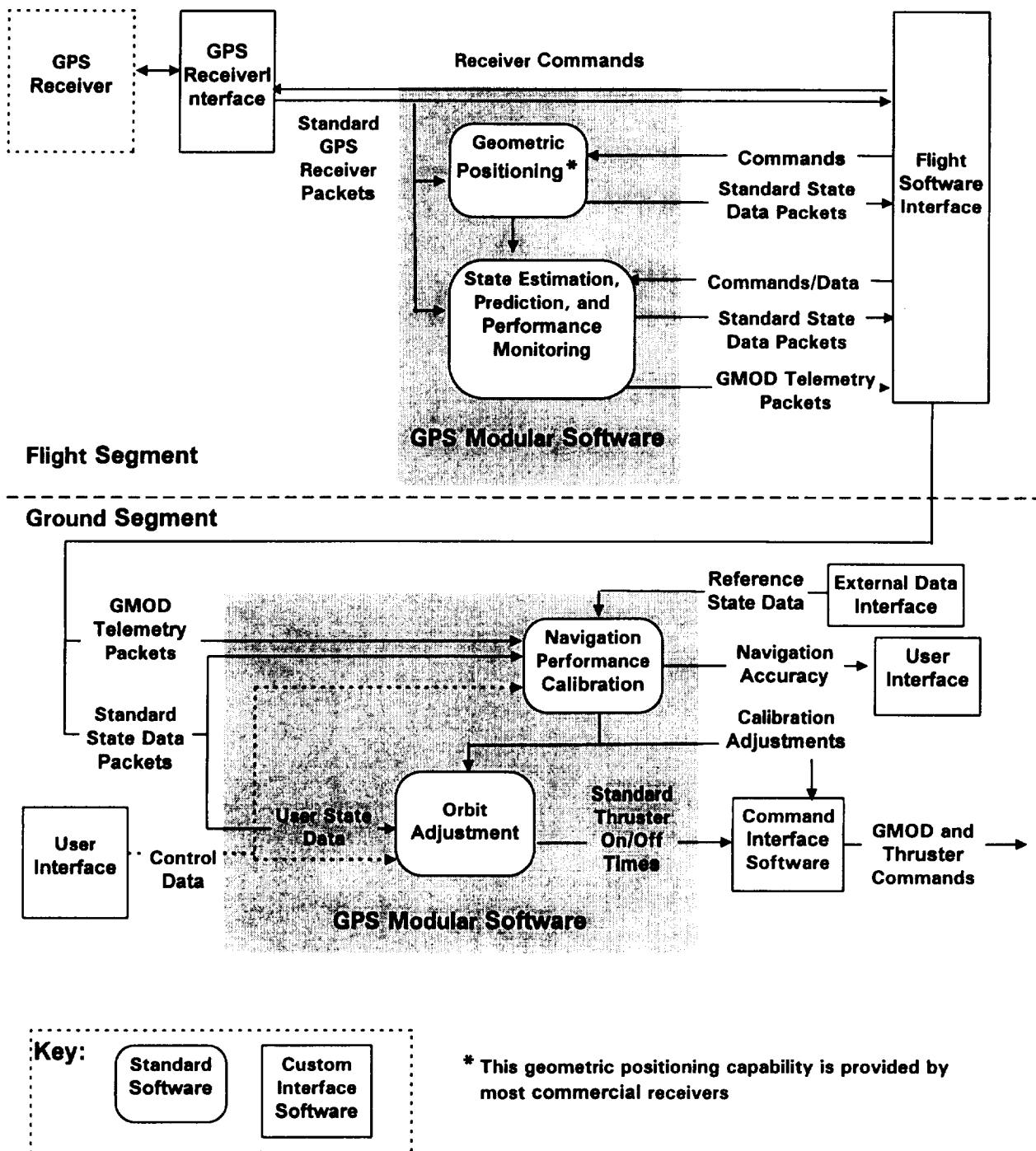


Figure 1. GPS Navigation Processing Components for Autonomous Navigation Configuration

GPS receiver interface software is hosted onboard to convert the output of the commercial GPS receiver (e.g., measurements, GPS broadcast messages, and geometric state estimates) to standard GPS receiver packet formats for downlink and/or use by GMOD and other spacecraft processes. The onboard navigation software functions can be hosted on a processor within the GPS receiver unit, on the spacecraft's onboard computer (OBC), or on a separate processor, whichever processor has sufficient available resources. The geometric positioning solution can optionally be used to provide an initial state vector for the filter processing, as well as to provide a real-time onboard performance check. Monitoring of the real-time performance of the state estimation process is also performed onboard, consistent with the MOCA goal of reducing the need for ground contacts to one per day. GMOD telemetry packets are provided at a commandable rate; these packets contain data needed for in-depth performance calibration and problem diagnosis.

The downlinked state estimates are propagated in ground software for use in the orbit adjustment computations and in the generation of predicted orbit product data. The ground-based GPS navigation calibration functions consist of reporting any available GPS health and receiver status information; verifying the accuracy of the spacecraft state estimation and orbit adjustment ground computations; and, optionally, verifying the quality of the GPS measurements and computing modeling adjustments to improve navigation performance. Calibration of the accuracy of the estimated spacecraft state can include comparison with a reference solution, such as one generated using another tracking system.

2.3 System Requirements

The GMOD context diagram, shown in Figure 2, illustrates the interfaces between GMOD and the other flight and ground software components. The GPS receiver interface software, which converts the output of the selected GPS receiver to standard GPS data interface formats, is specific to each receiver vendor/product. The flight system executive and spacecraft command, telemetry, and product data interfaces are specific to the flight environment in which GMOD is embedded. Similarly, the ground system executive, user control and display, product data, and external data interfaces are specific to the ground environment in which GMOD is embedded.

Functional requirements were defined for the following major GMOD functions:

- Determine geometric position
- Estimate user state
- Predict user state
- Monitor/calibrate GPS navigation performance

In addition, high-level operational and performance requirements were defined to support the four GPS operational configurations. These requirements are provided in Sections 3.2 through 3.4 of Reference 4, respectively.

3.0 GMOD Architecture and Design

This section discusses the definition of a candidate GMOD software architecture and a standard set of software applications that provide the recommended software partitioning between the flight and ground segments. Also given is a preliminary assessment of the feasibility of using a C++ based implementation for the GMOD software.

3.1 GMOD Software Architecture

The major design goals for the GMOD software architecture are the following:

- **Modularity and Configurability**—The software components that comprise the GMOD software must be able to be combined in a flexible manner to meet the required software configurations as defined in the GMOD operations concepts.
- **Reuse**—The design of the GMOD software must facilitate use of the same software components in both the flight and ground segments, to the extent possible.

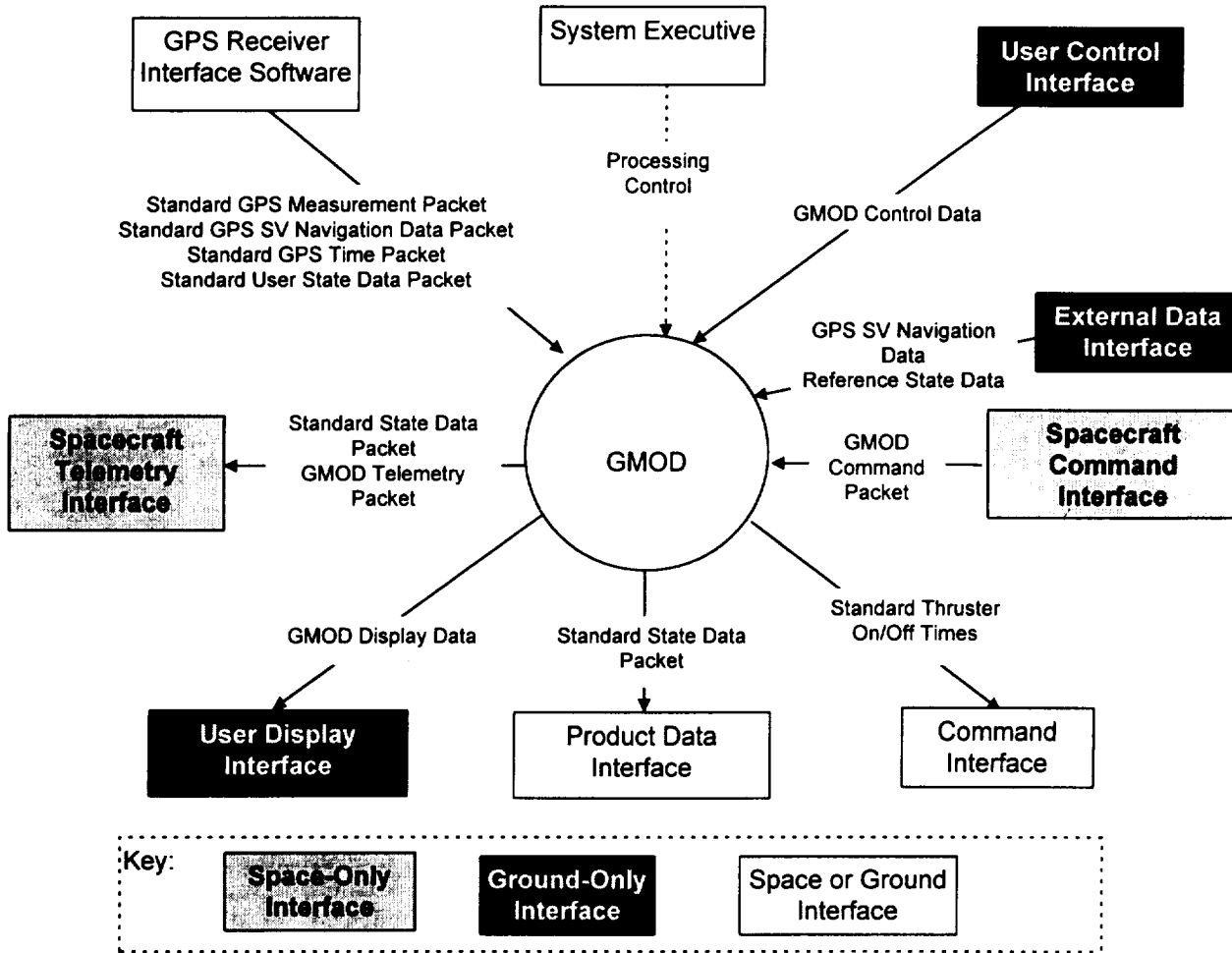


Figure 2. GMOD Interface Diagram

- **Encapsulation of Interfaces**—The GMOD software design should support isolation of the external interfaces (data and control) to the GMOD software to minimize the customization needed to interface the GMOD software with various commercial GPS receivers and spacecraft computer environments.

The selected GMOD software architecture depicted in Figure 3 meets each of these goals. The GMOD software architecture is based on the concepts developed for the GSFC FDD's Flight Dynamics Distributed System (FDDS) Generalized Support Software (GSS) for configuring application programs from a set of object-oriented, modular reusable components (classes) resident in a class library (Reference 10). The GSFC FDD has developed a set of software implementation standards and a generic software architecture for building reusable components for ground-based systems. The software implementation standards are referred to as the GSS implementation concepts. The GMOD architecture for the ground segment is based directly on FDDS/GSS, and the GMOD flight segment architecture extends the FDDS/GSS concepts to support the flight environment.

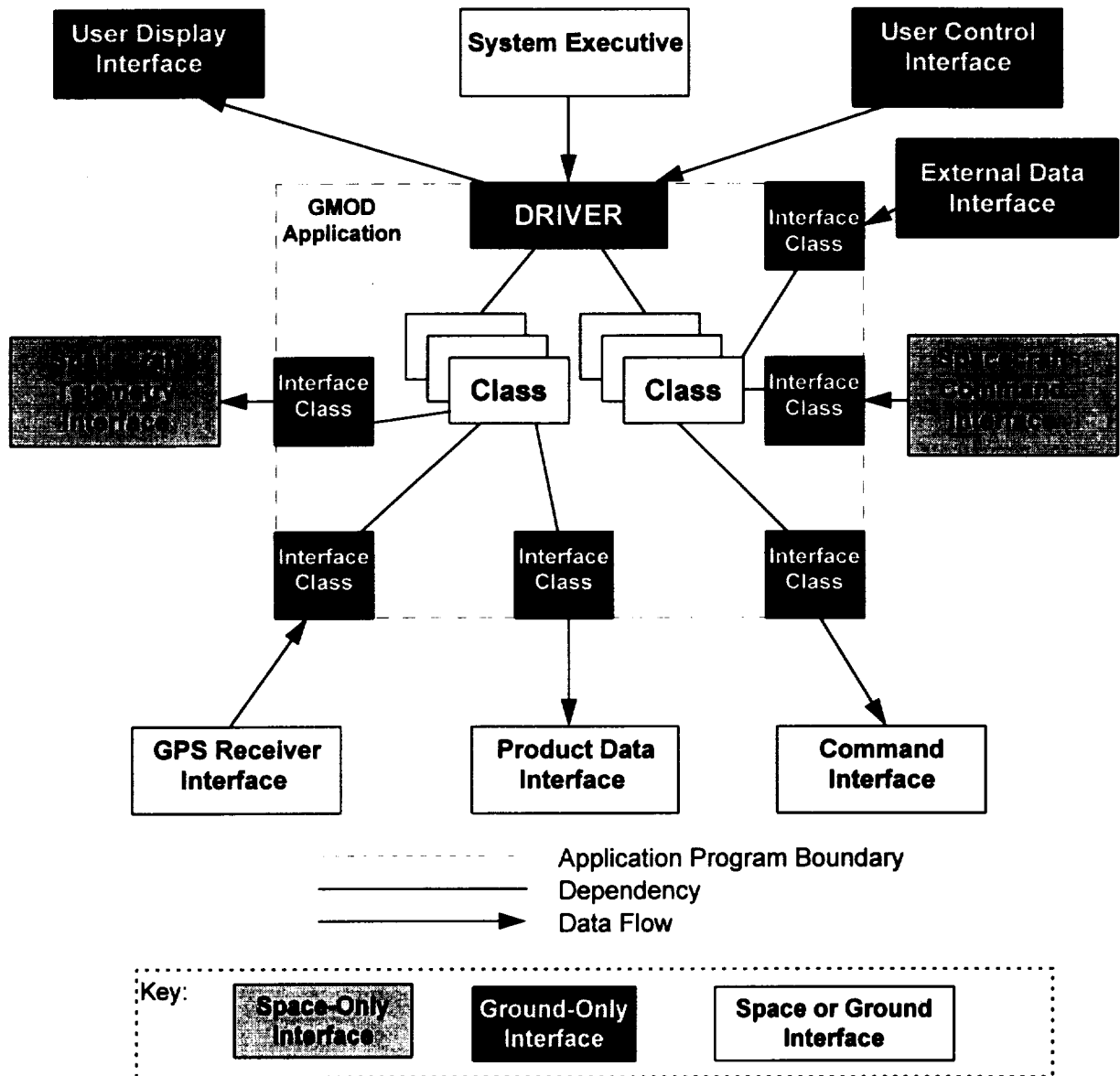


Figure 3. GMOD Software Architecture

As shown in Figure 3, the basic building blocks for a GMOD application are classes and drivers. A class is a software representation (abstraction) of an object in the GMOD problem domain of autonomous spacecraft navigation. Examples of objects in the GMOD problem domain are “physical” objects, such as the spacecraft, Earth, Sun, and Moon, and more abstract or “algorithmic” objects, such as integrators and measurement models. The shading of the boxes that represent the external interfaces to the GMOD application indicates whether the interface is ground only, space only, or both. The dark shaded boxes inside the dashed box that represents the GMOD application (labeled “Driver” and “Interface Class”) are those elements of the GMOD software architecture that may require some degree of customization for each mission-specific ground and space segment hardware/software configuration. The unshaded boxes (labeled “Class”) indicate those elements of the GMOD software that are reusable between specific mission ground and flight hardware configurations.

Classes are also used to encapsulate external data interfaces and associated access methods. As depicted in Figure 3, an interface class can be defined for the GPS receiver that contains the member functions that perform the data conversion needed to transform the receiver-specific data formats into the standard GMOD data format.

The driver element of the GMOD software architecture is the "glue" that binds the classes and data objects into an executable application (module). The driver element consists of an overall executive routine that manages the flow of execution in the configured application and a set of routines that manages the data exchange between the GMOD application and the user interface/executive (ground segment) or the flight software executive (flight segment).

3.2 Standard Software Components

The GMOD team also defined a set of standard applications that could be used to provide the functionality defined for each of the GMOD navigation configurations. Based on an analysis of the GMOD requirements and designs of existing FDD software systems that have similar requirements, the following six applications were defined:

- **Geometric Positioning**—Determines spacecraft position using the geometric positioning method; this function is typically resident in the software included in the GPS receiver unit by the GPS receiver manufacturer
- **Message Extraction**—Extracts the data from the GPS receiver custom data formats (onboard) or from the downlink telemetry stream (ground) and converts it to the GMOD standard data formats
- **Navigation**—Provides the autonomous navigation functions, including processing the GPS measurements and performing user spacecraft state estimation and prediction
- **Navigation Performance Evaluation**—Provides data analysis and calibration functions for the navigation application program
- **Orbit Adjustment**—Provides the autonomous orbit maintenance functions of planning, executing, and calibrating spacecraft orbit adjust maneuvers

Each of the applications listed above has the same software architecture as that depicted in Figure 3.

Section 4.2 of Reference 4 provides a preliminary list of software classes that comprise each of these applications.

3.3 Preliminary Feasibility Assessment

One of the ground rules given to the GMOD team at the start of the project was that the desired implementation language for GMOD was C++, because of its increasing use by the GSFC FDD in developing FDD's reusable software. To evaluate the use of C++ and also to evaluate the feasibility of using the FDDS/GSS software implementation standards (concepts), the GMOD team produced a small scale prototype of the Earth gravity model in C++. Based on the evaluation of the memory usage and central processing unit (CPU) usage, the GMOD team determined that there were no inherent problems with using C++ and the FDDS/GSS implementation standards for both the ground and space segments of GMOD. Prototyping of a larger application is currently in progress and will be evaluated prior to the start of full-scale implementation of the GMOD software. Section 4.3 of Reference 4 provides a detailed discussion of the associated feasibility issues.

3.4 Interface Standards

Data interface formats were defined for each of the GMOD software interfaces shown in Figure 3. These packet formats are defined so as to conform with the Consultative Committee on Space Data Standards (CCSDS) recommended packet structure defined in Reference 11. The CCSDS packet definition was adopted based on the recommendations of the MOCA program and because of its increasing use among NASA missions. Although the packet structures were designed to support the flight system interfaces, they should also be usable for the ground system user control and display data interfaces. A detailed description of each command and telemetry packet is provided in Section 5 of Reference 4.

4.0 Follow-On Activities

The initial GMOD study established the feasibility of developing modular software to support GPS navigation using commercial GPS SPS receivers. Beginning in July of 1996, the GSFC FDD will use the autonomous navigation configuration associated with the GEODE experiment hosted on the Small Satellite Technology Initiative (SSTI)/Lewis spacecraft (Reference 12) as an opportunity to assess the performance of the autonomous navigation configuration and the overall completeness of the GMOD capabilities to support this configuration.

Currently, the FDD is pursuing activities to broaden the scope of the initial study and demonstrate that modular software applications can be built that are suitable for both flight and ground environments. These activities include extending the domain of the GMOD concepts, requirements, interfaces, and software architecture to address impacts arising from the following items:

- Performance improvements to be gained through the use of Wide-Area Augmentation System (WAAS)-compatible GPS SPS receivers and GPS PPS receivers
- Expected performance for spacecraft with limited or intermittent GPS visibility, e.g., geosynchronous spacecraft, Space Shuttle, Space Station
- Improving the reliability of GPS autonomous navigation by using the GMOD autonomous navigation filter solution to reinitialize the GPS receiver
- Multiple spacecraft applications (e.g., formation flying and rendezvous)
- Very-high-accuracy differential GPS applications
- Detailed system requirements for autonomous orbit adjustment

In addition, the FDD is currently developing a prototype GMOD application to demonstrate the GMOD application development concept. The prototyping effort is divided into two phases. In the first phase, an autonomous navigation application is being developed to operate as a ground-based application. This phase will demonstrate that an autonomous navigation application can be built using a set of generalized C++ routines. This phase will also provide a benchmark against which the second phase prototype may be evaluated. The second phase will be to port the core classes of the first phase prototype into a typical flight software environment. The flight software environment will be provided by one of the GSFC MOD flight software test beds. This phase is intended to demonstrate that the software being developed to support GMOD ground applications can be reused in a flight environment with a minimal amount of modification.

Acknowledgments

The GMOD work was supported by NASA Headquarters (HQ)/Code OI and was a cooperative effort between personnel from the GSFC FDD and MOD, JSC, and JPL. The authors wish to thank the staff from GSFC, JSC, and JPL who contributed to this effort. Special acknowledgment is given to John Rush (NASA HQ/Code OI), Russell Carpenter (JSC), and Lawrence Young (JPL) for their support of this work. The authors acknowledge the contributions from D. Bolvin, J. Chang, C. Ewald, and J. Lorah of CSC and J. Pepper of GSFC.

References

1. Goddard Space Flight Center, *Final Report of the MOCA's GPS (MOCA/GPS) Splinter Team*, March 20, 1995
2. National Research Council, *The Global Positioning System A Shared National Asset*. Washington, DC: National Academy Press, 1995
3. M. Lichten et al., "An Automated Low-Earth Orbit Determination System with High-Accuracy Real-Time Capability," presented at the 1995 National Technical Meeting of the Institute of Navigation, Anaheim, California, January 18–20, 1995

4. Goddard Space Flight Center, Flight Dynamics Division, 553-FDD-95/014R0UD0, *Global Positioning System (GPS) Modular Software (GMOD), Feasibility Analysis Report*, D. Berry et al., prepared by Computer Sciences Corporation, September 1995
5. Goddard Space Flight Center, Flight Dynamics Division, 553-FDD-93/030R1UD0, *Tracking and Data Relay Satellite System (TDRSS) Onboard Navigation System (TONS) Flight Software Recommended Algorithms, Revision 1*, A. C. Long (CSC) et al., prepared by Computer Sciences Corporation, December 1993
6. Martin Marietta Corporation Astro-Space, Document No. 20045510, *Software Requirements Specification Flight Software—Navigation CSCI for EOS-AM Spacecraft (SD-110a)*, J. R. Herberg, November 7, 1994
7. Goddard Space Flight Center, Flight Dynamics Division, 553-FDD-95/011R0UD0, *Global Positioning System (GPS) Enhanced Orbit Determination Experiment (GEODE) Requirements and Functional Specifications, Draft*, A. C. Long (CSC) et al., prepared by Computer Sciences Corporation, May 1995
8. — — —, Flight Dynamics Division, 553-FDD-95/013R0UD0, *Global Positioning System (GPS) Enhanced Orbit Determination Experiment (GEODE) Mathematical Specifications, Draft*, T. Lee and A. C. Long (CSC), prepared by Computer Sciences Corporation, February 1995
9. — — —, Flight Dynamics Division, 553-FDD-94/007R1UD0, *Tracking and Data Relay Satellite System (TDRSS) Onboard Navigation System (TONS) Ground Support System (TGSS) Requirements and Functional Specifications, Revision 1*, G. Horstkamp (CSC) et al., prepared by Computer Sciences Corporation, July 1995
10. Goddard Space Flight Center, Flight Dynamics Division, 552-FDD-95/010R0UD0, *Flight Dynamics Distributed System (FDDS) Software Architecture, Volume 1, Draft*, D. Boland (CSC) et al., prepared by Computer Sciences Corporation, June 1995
11. Consultative Committee on Space Data Standards, CCSDS 102.0-B-3 Blue Book, Recommendation for Space Data System Standards, *Packet Telemetry*, November 1992
12. Frank H. Bauer et al., "The GPS Attitude Determination Flyer (GADFLY): A Space-Qualified GPS Attitude Receiver on the SSTI-Lewis Spacecraft," *ION GPS-95 Proceedings*, ION GPS-95 Conference, Palm Springs, California, September 1995

