A GENERIC OBJECT-ORIENTED DESIGN FOR A RADIO FREQUENCY SIMULATION IN A SPACE TELEMETRY AND COMMAND ENVIRONMENT

Joseph Policella
CAE-Link Corporation
2224 Bay Area Boulevard
Houston Texas 77058-2099

ABSTRACT

In a generic telemetry simulation the overall fidelity of the simulation is largely based on the simulated vehicle's On-Board-Systems (OBS) engineering models that drive the generation of the telemetry. Also, the actual transfer of data between the simulated vehicle and control center depends on the ability of the Radio Frequency (RF) OBS to acquire and process the RF links thus resulting in a Acquisition of Signal or Loss of Signal (AOS/LOS) determination. The simulated RF links are a function of the communications OBS models, and the communications environment models. The communications OBS models are responsible for propagating the RF signal. Since the RF link analysis is highly integrated into the characteristics of the communications equipment and environment models, RF link software needs to be constantly redeveloped as communications equipment models change, fidelity is added, or multiple links are created. However, by using a generic object-oriented design, RF link software can process any number of differing links based on the RF characteristics of the propagated wave. As a result, the communications equipment model software can be changed to reflect possible design changes without having to rewrite the RF link software thus allowing reuse of existing code.

INTRODUCTION

Accurate high fidelity simulations of RF links are necessary for high fidelity spacecraft telemetry simulations. Spacecraft telemetry simulations traditionally have been used to train ground crews, flight crews, and mission support personal in the use of OBS and ground telemetry equipment. As telemetry simulations evolve and become more affordable, other uses are being realized in areas such as RF link margin analysis, telemetry equipment design, telemetry equipment checkout, and mission planning.

The purpose of this paper is to demonstrate the advantages of an object-oriented approach as opposed to a functional approach for the simulation of RF links in a multi-vehicle space environment. A multi-vehicle space environment involves communications with Space Shuttle Orbiters, payloads, free-flyers, the Tracking and Data Relay Satellite System (TDRSS), the Ground Space Tracking Data Network (GSTDN), the Space Station Freedom (SSF), and other transmitting and receiving communications systems. For most space applications, users communicate with the vehicles through a ground station or a Tracking and Data Relay Satellite (TDRS). Currently, there are 3 TDRSs with each TDRS providing 1 Multiple Access (MA) and 4 Single Access (SA) forward services simultaneously along with 20 MA and 4 SA return services simultaneously. These services include S and K-band frequencies, various modulation schemes, and a host of other options. As a result, the number of possible combinations and type of links in a multi-vehicle space environment is immense.

This paper presents an object-oriented design which allows receivers to interface to transmitters in such a way that the transmitting system is transparent to the receiving system. This generic scheme allows all possible links to be modelled in the same manner with considerable reuse (see Figure 1). Each RF link in a point-to-point telemetry communications transmission is represented by instances of transmitter and receiver objects which are affected by environment objects.

OBJECT-ORIENTED VS. FUNCTIONAL

Non-object-oriented space telemetry communications designs have suffered in the past because the characteristics of the vehicles were deeply imbedded into the RF link processing software. Also, every possible link had to be modelled to ensure connectivity between the systems (i.e. a separate section of code existing for each link; TDRSS East to Orbiter, Orbiter to TDRSS East, TDRSS Spare to Orbiter, Orbiter to Ground Station, etc..). Therefore, when multiple systems were added the possible number of links increased geometrically (i.e. n systems yields $n \cdot (n-1)$ links), and within each link, the characteristics of the system were hard-coded into the link (i.e Ku-band, MA etc..). As a result, when a system changed or another was added, extensive rework was necessary.

Object-oriented space telemetry communications designs involve creating “classes” of real-world objects. These classes represent the attributes and operations of generic real-world objects. When classes are instantiated and assigned attributes specific to the vehicle being modelled, they become objects. By combining objects from several systems (i.e. environment, on board avionics, propulsion etc ...) a full vehicle can be modelled. As the vehicles being modelled evolve, vehicle models are easily updated by adding new objects, deleting obsolete objects, or by adding attributes and operations to existing objects. The various objects send messages to each other for information or services. As a result, when a real-world system changes, only the affected object needs to be updated, thus eliminating costly rework from systems containing heavily imbedded functional processes. For example, when vehicle B is added to Figure 1, a functional approach would require modification to both vehicles A and B.

RF LINK OBJECTS

The two main classes of objects for space telemetry communications are for the OBS communications equipment and the communications environment. The communications equipment class contains classes of antennas, RF Combiners, transmitters, and receivers. The Communications environment class contains classes of signals, ephemeris (noise), and obscuration (see Figure 2). Common to all communications equipment classes is the vehicle the equipment is contained in, and the operation of updating the equipment's mode by an on board computer or ground control command. Common to all communications environment classes is the coordinate system of the modelled universe. As a result, the subclasses of the communications equipment class and communications environment class are said to inherit the attributes and operations of their parent classes.

Communications Equipment Class - The basic subclasses of the communications equipment class are as follows:

1) The antenna class contains attributes for; polarization, the boresight angle, beam pattern, and gain. The operations include; updating the signal as its own attributes are changed due to moding, requesting a signal obscuration determination when receiving, and calculating the signal's gain based on boresight angles.

2) The RF combiner class determines which resources of the vehicle will update the signal and thus contains attributes for pointers representing object to object connections. On vehicles with multiple transmitters/receivers, antennas, signal amplifiers, and extensive cross-strapping, the RF combiner routes the signals to the various resources based on its commanded mode of operation. Also during reception, noise is routed to the receiver.

3) The transmitter class contains attributes for; the signal frequency, power, mode (modulation scheme), the presence of a Pseudo-Noise (PN) long code, PN short code (which uniquely identifies the vehicle), data rate, and if in dual channel mode, the I/Q channel power ratio. This information is set based on the commanded mode of operation of the transmitter. Basically, transmitters are responsible for creating (spawning), updating, and deleting signal objects.

4) The receiver class contains attributes for; the center frequency, acquisition bandwidth, acquisition threshold, mode (demodulation scheme), the presence of a PN long code, PN short code, detector lock, and receiver lock. Basically, receivers

are responsible for polling and processing signals. Processing includes; requesting availability from the RF combiner objects, calculation of the resultant signal power, calculation of a Bit Error Rate (BER), and from all this, calculation of the AOS/LOS determination. Also if AOS and tracking services are scheduled, the PN long code epoch is checked for coherent signals to allow ranging

Communications Environment Class - The basic subclasses of the communications environment class are as follows:

1) The signal class contains attributes for; the signal ID, polarization, frequency, power, propagation state vector, the presence of a PN long code, PN short code, data rate, and for dual channel applications, the I/Q channel power ratio. The operation of the signal class is to update its values based on messages received from antenna and transmitter objects.

2) The ephemeris class contains attributes for; the type, mean, variance, and level of the noise present in the simulation environment. The noise parameters are used by receiver objects to determine the Signal to Noise Ratio (SNR) which is used for calculating the BER and AOS/LOS determination. The ephemeris class updates its values usually from instructor inputs or other environment objects.

3) The obscuration object contains attributes for the position and size of an obscuring body. The operations include; calculating whether the receiving antenna is pointed at the sun within a given tolerance thus causing outage, and determining if the body in the simulation is blocking the signal's path to the receiving antenna. This class is instantiated to produce obscuration effects due to the earth, sun, and any payloads that may be in the environment.

When higher fidelity is required to model the real-world telemetry equipment of the vehicle or environment, extra classes are created and inserted under the appropriate class. Also, more objects can be added by instantiating the desired class and forming objects from other objects (aggregation), allowing representation of a variety of vehicles and real-world objects (see Figure 3). As a result, the development environment is flexible and can represent many dynamic configurations.

PROCESSING THE COMMUNICATIONS PATH

The processing of an RF link involves utilizing all the objects in the communications path. Each active receiver must process each active signal. In the real-world, many transmitters are simultaneously active causing the RF environment to contain many signals and noise. Also, many receivers constantly search the RF environment for a compatible signal to lock on. As a result, in a real-world RF link, the transmitter is totally transparent to the receiver and the intended communications path can be any combination of active transmitters and receivers.

To accurately simulate this process, several parallel processes must occur. First, a signal is created by a transmitter. Next, an RF combiner routes the signal to the appropriate antenna. Then, the antenna updates the polarization and power attributes of the signal based on its abilities. At this point, the signal is placed in a stack with all other active signals. While waiting to be picked up by a receiver, the signal is constantly being updated by the transmitter and antenna objects as their configurations change. In a separate task, a receiver starts to poll signals when it is moded to do so. The receiver polls signals by checking the PN short code of each active signal. If a match is found, the receiver requests that the signal be made available by the RF combiner and antenna objects. This causes the RF combiner object to send a message containing the signal ID to the linked antenna object. The antenna object then rendezvous with the obscuration object to determine if the signal can be seen by the antenna. If it can, the signal object's attributes are updated based on the antennas gain and boresight. Finally, the signal and the noise currently present in the environment are routed to the receiver, where the final AOS/LOS determination is made based on the attributes of the signal, the noise, and the receiver (see Figure 4)

With the advent of multi-processor hosts, the real-world RF environment can be accurately simulated by placing each active receiver, signal, and transmitter in its own processor. This allows simultaneous processing of links involving many possible combinations of receivers and transmitters in real-time. As a result, signals are processed with the transmitter completely transparent to the receiver, whereas a functional approach must define the receiving and transmitting pair to know which type of link to process. Also, the signal objects are dynamic. This causes system resources to be allocated and deallocated as the signals are created and deleted, thus maximizing system efficiency.

CONCLUSION

Previous functional RF link designs, while well suited for initial development, became quickly obsolete as telemetry equipment and transmission methods evolved and became more complex. As a result, the simulation software which modelled the real-world telemetry equipment and processes was constantly being updated. Since, the functional approach did not lend itself easily to modifications, it became expensive to keep the simulation current with telemetry technology. However, an object-oriented RF link design overcomes many of these shortcomings by representing the real-world objects with simulated objects. With instantiation and inheritance, large models can be quickly prototyped and modified due to reuse. The simulation also can be integrated with other simulators easily because an addition to the communications path causes generic classes to be instantiated and not re-engineered. As a result, an object-oriented RF link allows the simulation to be co-developed with the real-world vehicle and delivered to the user before the real-world vehicle is built, thus allowing design validation, checkout, and training prior to the vehicle's deployment.

ACKNOWLEDGEMENTS

The author wishes to thank William D. Wessale for providing his expertise in Object-Oriented Design techniques. Also, significant contributions were provided by Howard W. Stein and David J. Retherford.

REFERENCES

Coad, Peter and Yourdon, Edward, Object-Oriented Analysis, Prentice-Hall, Inc Englewood Cliffs, New Jersey, 1990.

Rumbaugh J, Blaha M, Premerlani W, Eddy F, and Lorensen W, Object-Oriented Modeling and Design, Prentice-Hall, Inc Englewood Cliffs, Hew Jersey, 1991.

Booch, Grady, Software Components with Ada, Benjamin/Cummings, Menlo Park, California, 1987.

NASA, STDN 101.2 "Space Network (SN) Users' Guide Rev 6", Goddard Space Flight Center, Greenbelt, Maryland, Sept 1988.

NASA, JSC-19754 "Shuttle Mission Simulator Program Maintenance Manual, Acquisition of Signal/Loss of Signal Software Subsystem XA", Lyndon B. Johnson Space Center, Houston, Texas, 1984.

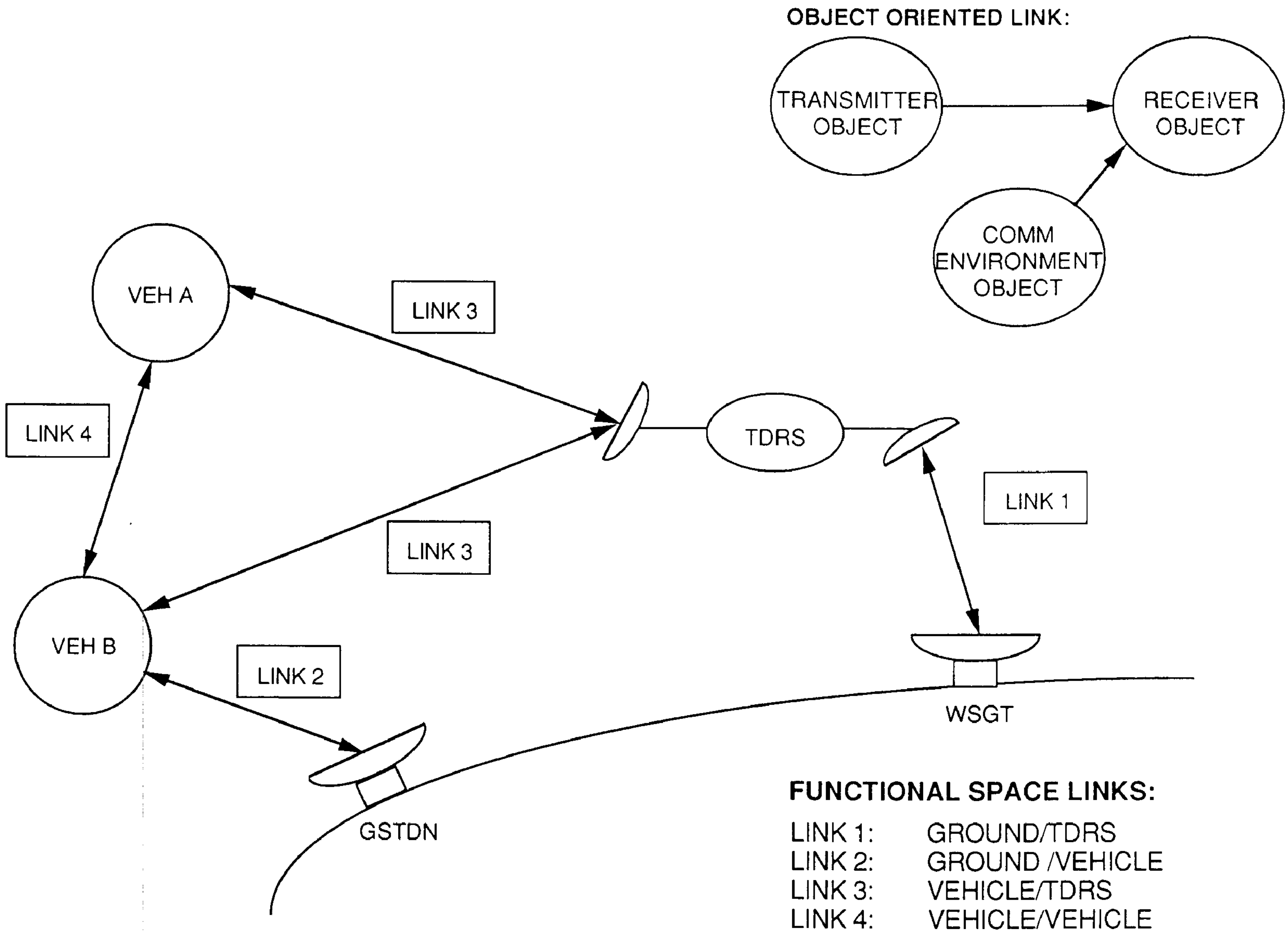

FIGURE 1: OBJECT ORIENTED LINKS AND FUNCTIONAL LINKS

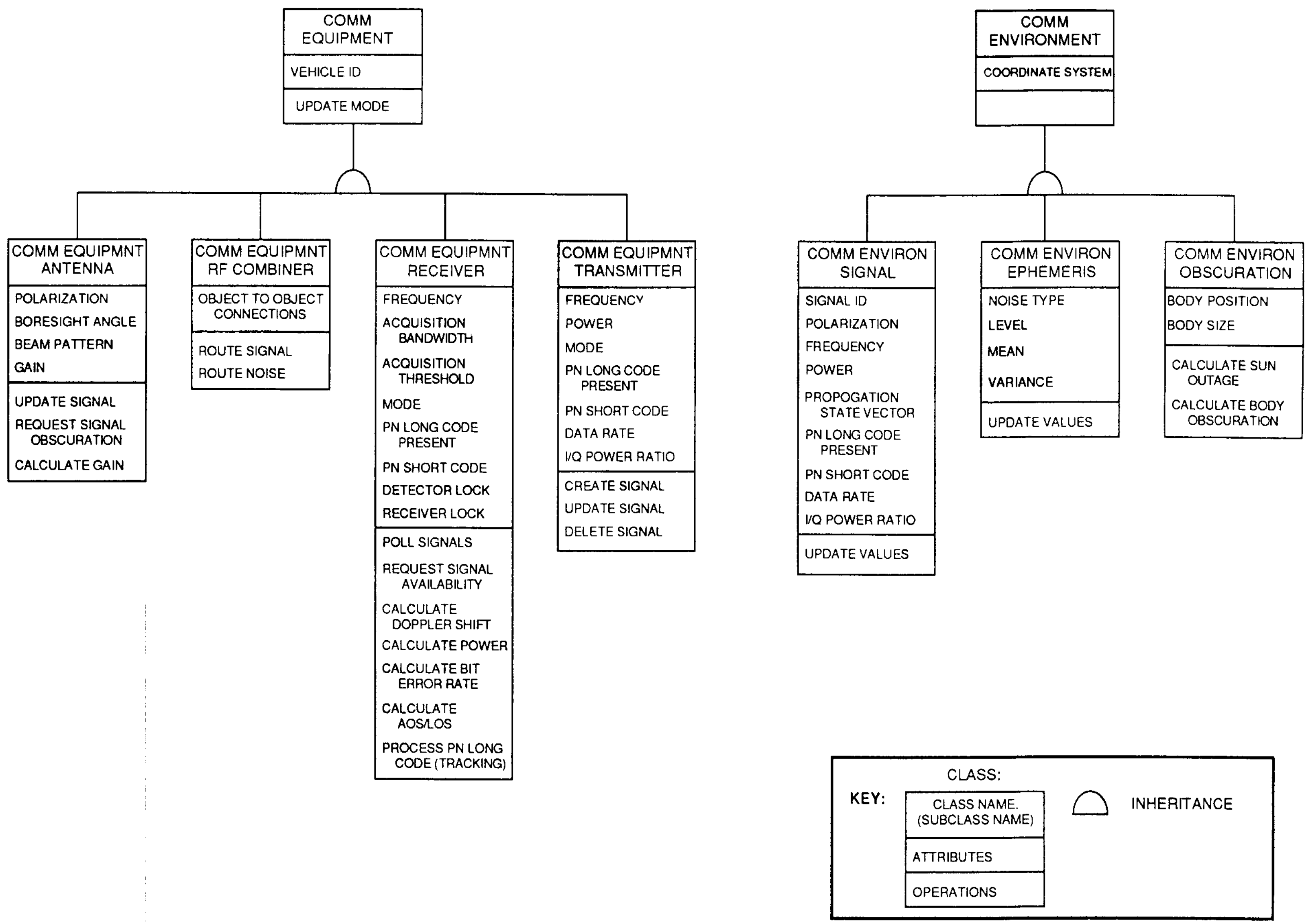

FIGURE 2: COMMUNICATIONS EQUIPMENT CLASSES AND COMMUNICATIONS ENVIRONMENT CLASSES

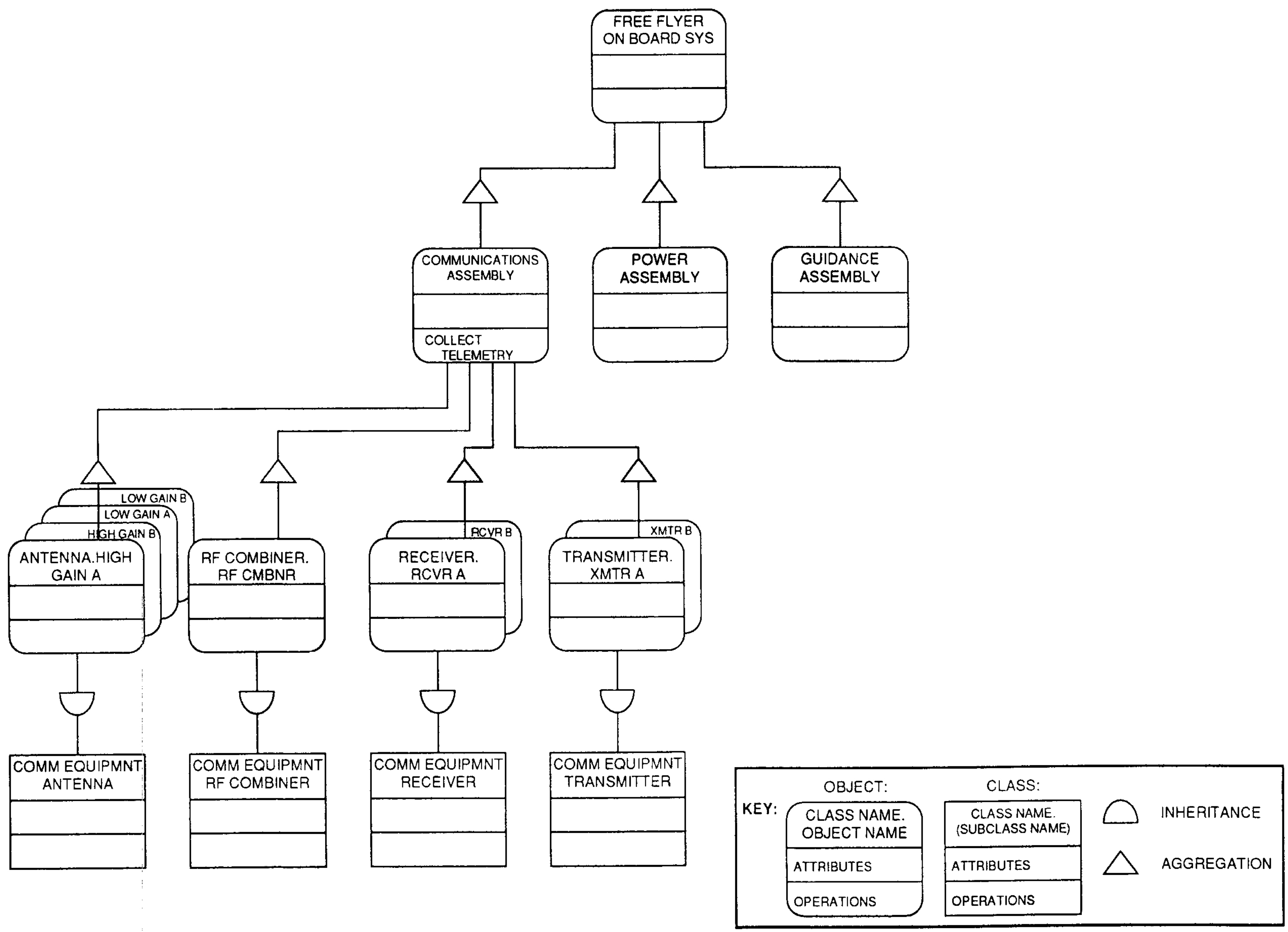

FIGURE 3: EXAMPLE OBJECTS CREATED BY INSTANCIATION AND AGGREGATION

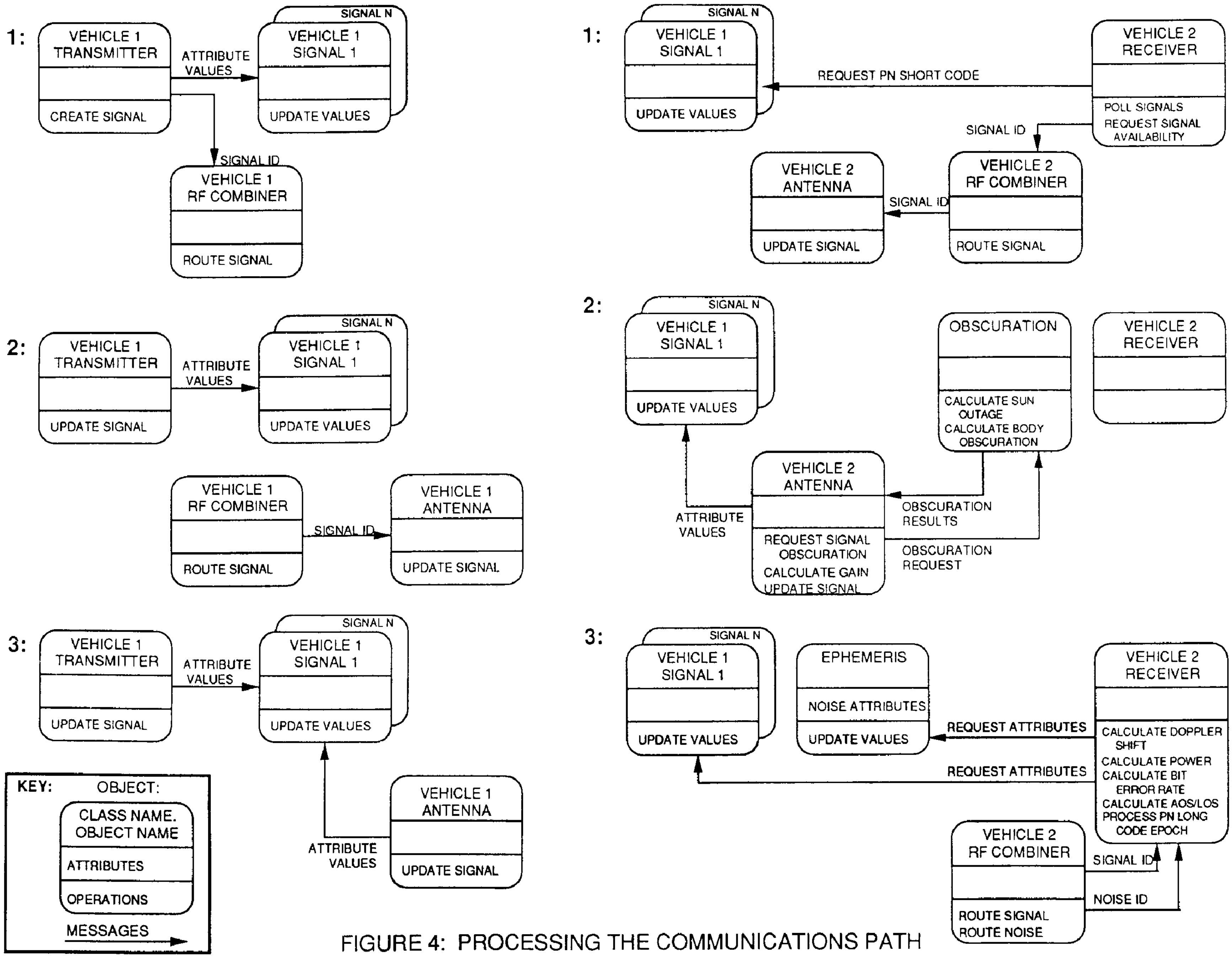

FIGURE 4: PROCESSING THE COMMUNICATIONS PATH