STREAMLINING SATELLITE DEVELOPMENT, TESTING, AND OPERATIONS USING A COTS COMMAND AND TELEMETRY PACKAGE

Andrew W. Lewin
Orbital Sciences Corporation
Dulles, Virginia

Abstract

This paper discusses how modern, commercial off-the-shelf (COTS) command and telemetry packages and data analysis tools can reduce cost and tighten schedules. By including the ground station from the start of the program life cycle, organizations can improve efficiency through the parallelization of activities, reuse of program resources, improvement of processes, and elimination of development tasks. The benefits of this approach are substantial for a single program. However, corporations can realize even greater productivity when subsequent programs build upon the strong foundation created by their predecessors. The methods introduced in this paper are brought to life through the example of the development of the ORBCOMM constellation of small communication satellites.

Introduction

Customer cost sensitivity, coupled with increasing competition in the commercial satellite production market, are placing ever-increasing pressure on aerospace companies to simultaneously reduce both cost and schedule. In this environment, ground station development often receives little attention until the last possible moment. This decision overlooks a tremendous opportunity for improving program efficiency. Inexpensive, commercial off-the-shelf (COTS) ground station tools are available which can streamline the spacecraft development process from the proposal phase through on-orbit operations.

Process improvements start with the ground station itself. Powerful, yet easy to develop, command and telemetry packages are now available at a minimal cost. These tools enable rapid ground station development and quick response to changing spacecraft designs. Such technology improvements make it possible to integrate the ground station into the development program at a very early stage. This capability is valuable, since operational ground station functionality substantially overlaps that required for other aspects of a spacecraft development program, especially integration and test. In addition, ground system development has synergy with other aspects of the development process.

The application of these concepts is described for the development of the ORBCOMM constellation of small communication satellites. Following the launch of the first two spacecraft in April 1995, this aggressive program calls for the construction of 34 flight spacecraft and several ground test vehicles. Programmatic challenges include an 18-month redesign/development to launch time for the first two spacecraft and six weeks from "pieces to product"—the time elapsed from component delivery to a vehicle ready to be shipped to the launch site—for the bulk of the constellation production.

Many of the concepts discussed in this paper were successfully applied to the development of the original ORBCOMM vehicles. The full application of these methods has led to faster and lower cost development for the constellation spacecraft.

Reducing Life-Cycle Cost and Schedule

Before describing how a ground station can have such a substantial impact on satellite development programs, it is important to examine the more general problem of life-cycle cost and schedule improvement. This provides a framework in which to describe the benefits provided by command and telemetry tools.

Improving either cost or schedule is not especially difficult. Cost and schedule can be traded with relative ease. For example, extra shifts or personnel can be added to a program to make up schedule. However, simultaneously reducing both requires at least one of the four efficiency improvement factors:

- **Parallelization** of activities
- **Reuse** of hardware, software, or other program resources
- **Process improvement**
- **Elimination** of tasks

The impact of parallelization is the most complex of the four efficiency improvement options. Assuming no time value of resources, changing two serial tasks to parallel produces a schedule improvement with no cost impact. However, the "time is money" credo is operative on virtually all programs: the longer the program, the more it will cost. This result derives from intangible benefits of faster development such as corporate image, to quantifiable benefits such as earlier progress payments. Schedule is particularly valuable on commercial satellite programs, where earlier delivery means earlier revenue generation. Therefore, schedule reductions can actually result in automatic cost reduction as well.

Parallelization can also decrease both cost and schedule by taking advantage of synergy between tasks that are parallelized. For example, using the flight ground system for integration and test allows the ground system developers to help test the spacecraft hardware and software while the integration and test team aids in ground station debugging.

The reuse of resources either within a program or from program to program improves cost by allowing development or acquisition costs to be spread across multiple programs or project groups. At the same time, reuse improves schedule by eliminating unnecessary development tasks.

Process improvement is simply finding a better way of performing a particular task. Anything that improves the efficiency of an existing task or replaces a process with a better one falls in this category. Process improvement reduces schedule and cost by decreasing the amount of time and money it takes to complete a task.

Finally, task elimination produces its benefits for obvious reasons. This factor can be viewed as a subset of both reuse and process improvement. Reuse eliminates recurring development efforts, and process improvement often results from designing away parts of a task.

Proposals, System Design, and Rapid Prototyping

One of the greatest advantages of modern command and telemetry systems are their rapid prototyping capabilities. This is especially valuable during the early evolution of a satellite concept. At this stage, technical risk can often be mitigated by performing a partial system simulation. A command and telemetry package can be integrated with such an effort within a matter of days.

Even crude system simulations can be invaluable marketing tools for selling a satellite concept. The impressive graphics capabilities available in modern ground systems allow vendors to make a strong impact with a minimal development effort. The ORBCOMM command and telemetry system development process that makes this possible is shown in Figure 1.

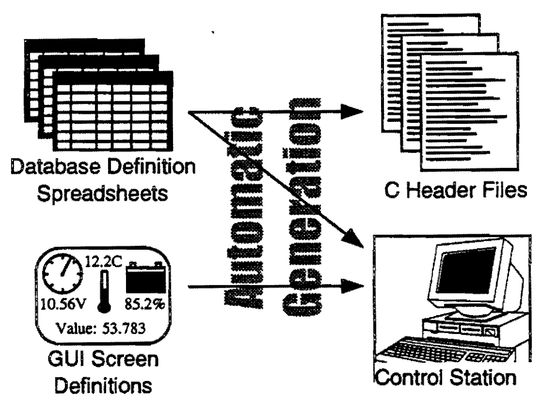

Figure 1. Ground Station Development Process.

The ground system employed by ORBCOMM is comprised of two primary user-specified components, the ground station database and the command and telemetry screens. The database tables describe all aspects of a command and control application such as telemetry decommutation, command generation, limit checking, and telemetry display. The screen definitions for the graphical user interface (GUI) are used to define the look and feel of the ground system. This includes both what types of objects are displayed and the actions to be performed when they are selected by an operator.

The keys to the rapid prototyping capability are the ease-of-use of the definition tools and the automated generation of the final products. For

ORBCOMM, the command and telemetry packet definitions are specified in Microsoft Excel. The spreadsheet columns include information such as command names, telemetry packet identifiers, command parameter and telemetry point names and sizes, conversion factors, and command parameter and telemetry point limits. A custom-built database generator takes this input and creates the ground station database table entries. The input to the database generator is simply a tab-delimited text file, so any spreadsheet or database product can be used. By relying upon commonly available tools, the database specification process can be performed with existing software and without having to train personnel in the use of a new and unfamiliar system.

By avoiding software purchases and training costs, a program can clearly save money. However, this is not the greatest source of savings. If the database specification tool requires special training or is not readily accessible and easy to use, engineers will avoid it. For example, software engineers will modify the header files directly rather than change the spreadsheets and regenerate them. Inevitably this costs more time than it saves, such as when a critical path test is delayed while the command and telemetry system is modified to match the data structures manually edited by a software engineer.

The screen definition file is used to create the C code that handles all user input. This code reads the screen definition at execution time. Several sample command and telemetry screens and an integration and test automated script execution screen are shown in Figure 2. The generated C code is linked with user-defined C code and the ground station object files to create the final executable code. All of these capabilities are provided as part of the COTS ground station tool used by ORBCOMM.

On ORBCOMM, one of the useful by-products of the command and telemetry spreadsheets is automatically generated header files for the on-board software. The header files contain the command and telemetry data structure definitions and initialization parameters. These files are created by custom-developed macros written in Visual BASIC. This approach provides the standard benefits of automated processes: computers can write code much faster and with fewer errors than a human. However, the primary gains appear in integration and test. Early in the development cycle, the speed of the automatic

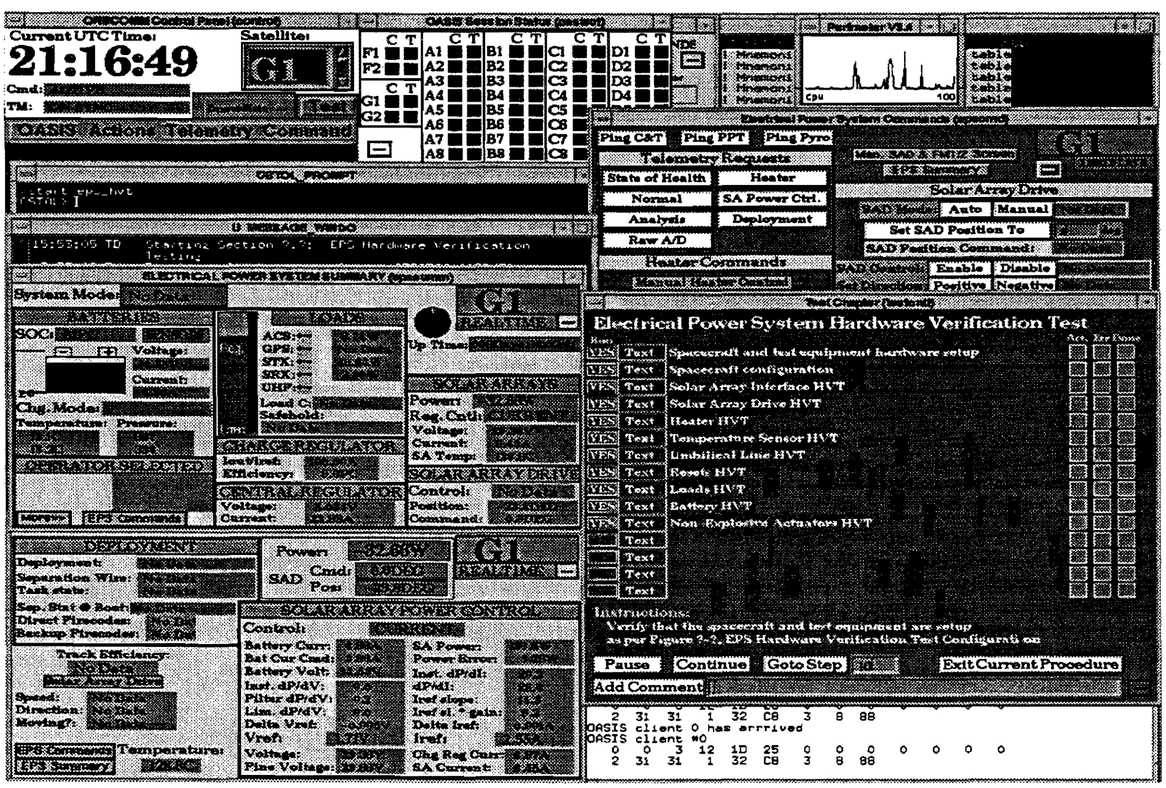

Figure 2. Sample Command and Telemetry and I&T Test Execution Screens.

generation process allows both the ground station and flight software developers to easily keep pace with frequently changing designs. Second, by using the same source, engineers ensure that ground station and flight software are always compatible. This eliminates the need for a dedicated verification effort late in the integration and test flow.

Integration and Test

The use of an advanced command and telemetry (C&T) system during integration and test (I&T) provides two major benefits, automation and shared development with flight operations. The rapid development capabilities and synergy with software development are applicable here as well. In addition, this use of the ground station supports parallelization of the test procedure development effort with software and ground station development. This is demonstrated in the partial program flow depicted in Figure 3.The program flow in Figure 3 is intended to provide a rough temporal illustration of a spacecraft development plan. At this level of detail, it is difficult to adequately demonstrate the benefits of the early use of the ground station in the program flow. One of the key points is the location of test script development in parallel with hardware development. This minimizes the amount of satellite test development performed in the program critical path. Thanks to rapid development capabilities, the ground station is never a threat to enter the critical path. This ensures that the critical path lies either in the software or hardware development where it belongs.

Few schedules are likely to indicate a large block of time in the critical path devoted to test script development and debugging. Nonetheless, in practice this is exactly the type of activity that can sneak into the critical path and create schedule slips. However, with an integrated development approach among ground station, software, and test procedure developers, it is truly possible to completely remove spacecraft test script development and nearly eliminate test script debugging from the program critical path, even with completely automated test scripts.

One of the greatest benefits provided by command and telemetry systems is the ability to automate spacecraft testing. Test automation provides a number of distinct advantages over manual testing. First, a computer can execute a test script much faster than a human can run a paper procedure. Second, automated tests are run the same way each time. This removes the variability among test conductors and ensures that procedures followed for each test are identical. Third, automated tests are much safer because they substantially reduce the risk of operator error. Finally, automated scripts can be executed at any time of any day, without the need for human oversight. This allows a program to make the maximum use of its most valuable resource: spacecraft hardware.

One must, of course, be cautious when operating a spacecraft unattended; however, ground stations provide endless possibilities for spacecraft monitoring. If the ground system software detects an anomalous condition, it can immediately stop the test and safe the vehicle. Furthermore,

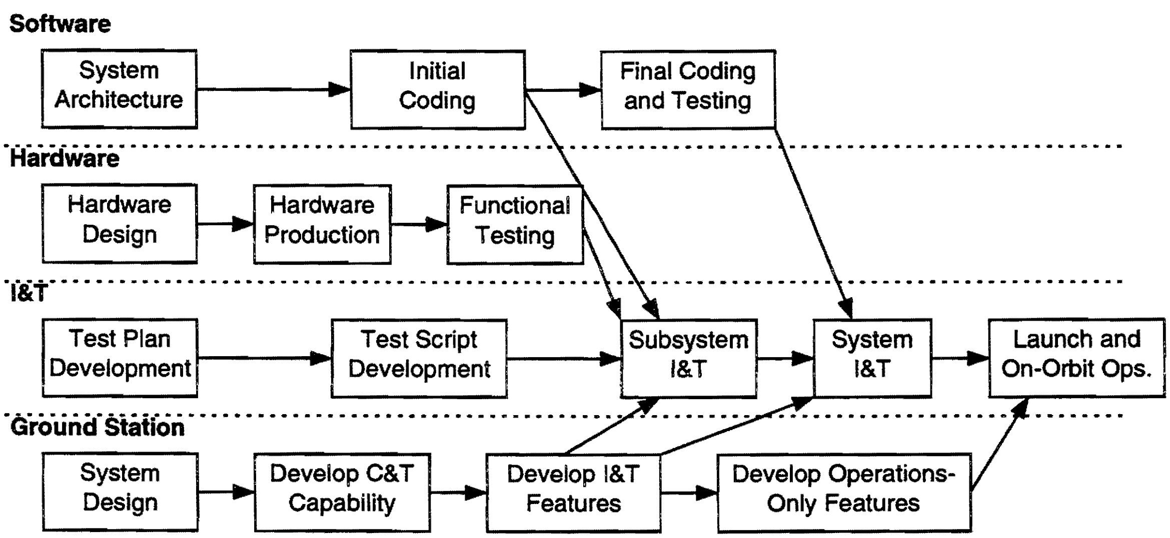

Figure 3. Partial Satellite Program Flow.

spacecraft monitoring functions developed for integration and test can also be applied to on-orbit operations.Generally speaking, one can expect developing automated test procedures to involve a greater initial investment than paper procedures. However, based upon ORBCOMM experience, the difference in development time is minimal. Furthermore, as the program flow indicates, this procedure development can be accomplished in parallel with hardware and software design, out of the critical path. The slight initial investment is repaid many times over as spacecraft integration and test progresses.

When developing a constellation of 36 spacecraft, the benefits of automation are obvious. However, automation is just as essential to single-spacecraft development programs. Even if only one spacecraft is flown, a qualification vehicle and possibly several development vehicles are also likely to be built. In addition, similar tests flow from unit to subsystem to system tests, so test development can be re-used. Finally, test repetition is inevitable. Ideally, identical tests are performed before, during, and after environmental testing. If failures occur during tests, procedures must be rerun. All of these factors combine to ensure that test automation will save every program time and money.

Test automation need not be limited to spacecraft commanding. The same features that make a command and control system useful for spacecraft commanding can be applied to all aspects of the system test environment. By automating the entire test environment, test setup time can be dramatically reduced. The use of the ground station on the ORBCOMM program is shown in Figure 4.In addition to commanding the spacecraft (performed via the data server), the ground station provides command and control over the entire test environment. In order to maximize flexibility, each of the tools is designed to be operated either in a stand-alone mode or controlled from the ground station. The test racks include equipment such as a solar array simulator, power supplies, dynamic load, modems, and antenna simulator. The racks are connected to the Ethernet via an Ethernet to GPIB converter to allow for external control over the rack instruments.

Most of the rack hardware is controlled from a commercially available product called LabView developed by National Instruments. Although the ground station is capable of commanding the test equipment, LabView is specifically designed for this task. In fact, LabView controllers are freely available on the Internet for most test equipment. Therefore, the only remaining task is to build in LabView a command and telemetry interface with the ground station. Once it is developed, this capability can be used on all subsequent programs.

In order to convince the satellite that it is in space, the attitude control system sensor signals must be replaced with values generated by the world environment simulation. In addition, the electrical power subsystem must be fed with an appropriate solar array input which also must be

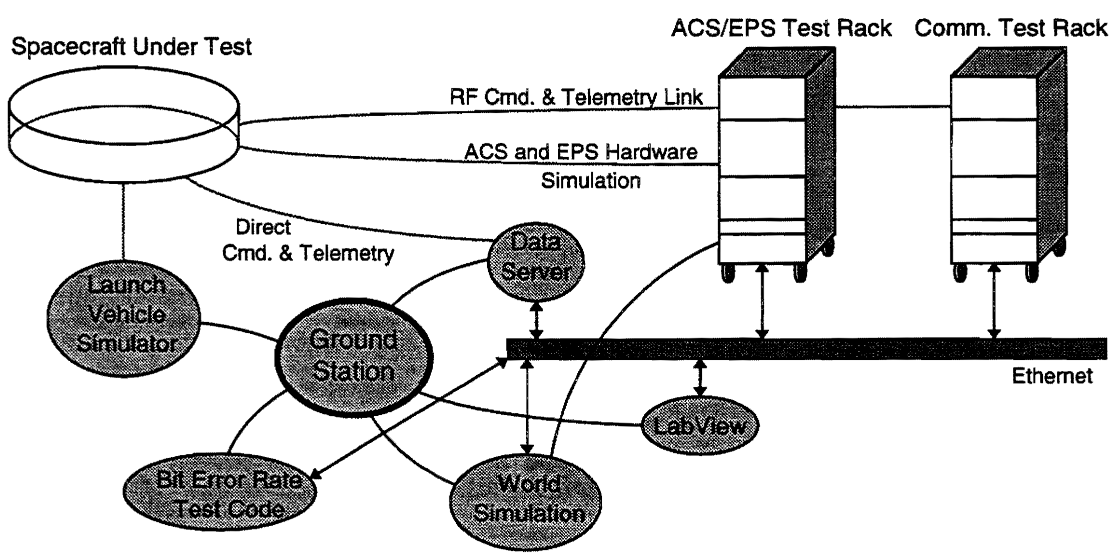

Figure 4. ORBCOMM Spacecraft Test Environment.

calculated by the simulation. The ground station interaction with the world simulation can be as simple as initiating a pre-programmed simulation. However, the simulation also provides the capability to manually command a specific operating environment.

Bit error rate testing of the ORBCOMM transmitters and receivers is performed with custom-developed software. Ground commanding is primarily limited to configuring the software and initiating and halting tests. This also holds true for the launch vehicle simulation software. It is used to test the interaction between the launch vehicle flight computer and the spacecraft during launch and deployment.

Although some of the software elements require little commanding, connecting them to the ground station provides other advantages. One of the greatest benefits is the handling of telemetry. Without this capability, each tool must log its own data and provide its own facilities for reviewing that data. On the other hand, the ground station can handle test environment telemetry with the same tools it uses to process satellite telemetry; no additional development is necessary. Furthermore, since the same tools and formats are used, an engineer can easily compare spacecraft telemetry and test environment telemetry, which can be invaluable when investigating test anomalies.

Using the eventual flight ground station has a second major advantage. The need for command and control of a spacecraft (or spacecraft subsystem) undergoing functional testing is nearly identical to that required for on-orbit operations. Therefore, a program needs only to invest in developing a single system for both applications. The box below lists just a few of the capabilities required for both on-orbit operations and integration and test.

<u>Ground Station Requirements Common To On-Orbit Operations and Integration and Test</u>

- Telemetry decommutation and conversion
- Telemetry display
- Limit checking
- Spacecraft health monitoring
- Command generation
- Command execution verification
- Message and data logging
- Data archiving
- Data analysis
- Script generation and execution
- Operator training

In addition to common functional requirements, the development schedule for integration and test and on-orbit operations are complimentary. Development of the initial command and telemetry screens can occur in parallel with initial software development and hardware design. Test developers can begin debugging test scripts even before hardware is available. Once development units arrive, final test script debugging and hardware checkout can commence.

By that time, an initial telemetry logging, message logging, archiving, and data analysis system must be in place. Although this may seem challenging, it is only an issue for the first program to use the command and telemetry system. Subsequent programs can use the same recording, archiving, and data analysis systems. Here again the command and telemetry tool can lend a hand with built-in telemetry and message logging capabilities.

While subsystem and unit testing progresses, ground station development heads towards completing the full set of command and telemetry screens. Spacecraft health monitoring functions critical to integration and test are implemented as well. Finally, command and control of test software such as the world simulation is developed and tested. This ensures that the ground station is prepared for satellite system testing.

By the time the spacecraft begins system-level testing, ground station development required for integration and testing is complete. At this stage, the ground station development team can devote its attention to the additional requirements of on-orbit operations.

At each step of the development cycle, the ground station is ahead of the requirements for integration and test. This may seem like a description of a program manager's utopia, but the capabilities of modern command and telemetry systems and data analysis tools make this much more of a reality than a dream.

On-Orbit Operation

When struggling to meet a difficult launch deadline, on-orbit operations may seem to be the least of a program's concerns, even if the developing organization is also responsible for satellite operations. However, many spacecraft are operational for longer than they spend in development. In fact, it is not uncommon for satellites to be turned off with useful life remaining simply to save the cost of continued operations.

One solution to this problem is "lights out", or totally automated operation. In this mode, operators only monitor some satellite passes. For the remaining contacts, the ground station follows an automated pass procedure. This includes downlinking satellite telemetry and uplinking command loads.

Lights out operation can take one of three increasingly automated approaches to health monitoring. The first method uses minimal health checking on the data received during the pass since no one is present to respond to the problem. Spacecraft health monitoring is performed with non-real-time data analysis tools during business hours.

The second approach uses the ground station to identify critical problems and notify an on-call engineer. Using current wireless technology, the ground station can automatically page an engineer when it identifies a serious problem. If further information is required, the command and telemetry package can generate reports which are then sent by fax or e-mail. At this point, the engineer can either go to the operations center in person or dial up the center and respond to the problem from a computer at home. Regardless of the response approach, this method requires additional investment in the ground station to define the conditions that trigger an alarm.

The most complex approach is to build the ground station with both automated anomaly detection and anomaly response. Implementing the automation is not the difficulty with this method. Rather, the majority of the time is spent defining what constitutes a problem and how to respond. However, this situation is analogous to the problem of test script development discussed in the previous section. If one is already making the engineering effort to define anomalies and the appropriate responses, it is nearly as easy to implement them as automated procedures as to create and maintain them in paper form.

The concept of lights out operation can be applied to data analysis as well. The ground station can be built to initiate a daily data analysis routine that performs a broad range of telemetry analyses and produces a summary report for operations staff. However, automated health checking cannot completely eliminate the need for data analysis by humans. Rather, a good automated telemetry evaluation system is complementary to engineering staff. The automated system performs relatively simple checks on most or all of the telemetry points, while a human performs the more difficult tasks of trending, complex data analysis, and subtle anomaly detection.

Again, the cost/benefit ratio of ground station automation must be considered. For a constellation of 36 low-Earth orbiting satellites the solution is obvious. With an average of one satellite pass starting every 4 minutes and as many as 10 satellites in view at one time over the United States alone, ORBCOMM cannot possibly afford to monitor every pass in its entirety. Therefore, a degree of ground station automation is essential. However, this level of activity can cost-effectively support at least one operations staff member around the clock. Therefore, the ORBCOMM ground station development is following the second automation option.[1]

In addition to standard limit checking, the ORBCOMM ground station is designed to look for problems such as missed satellite passes or units not sending telemetry. For critical parameters, the system includes more complex limit checking capabilities. For example, the battery voltage is evaluated as a complex function of the battery state of charge and the battery current. If an anomaly is detected, an error window is displayed to notify the operator. Real-time data analysis is augmented with an extensive automated daily data analysis routine that adds the necessary breadth and depth to the overall spacecraft health monitoring scheme.

Single satellite programs can also benefit from automation. Both the first and third automation options are logical choices. Clearly, the first option requires little ground station investment. The third option is also attractive. As previously mentioned, if the program had already intended to develop contingency procedures, only a minimal additional investment is required to create them in an automated rather than paper form.

Reuse

Up to this point, the discussion has focused on the benefits of ground station development to a single program. However, this is only part of the picture. Much of the investment made in the ground station can be applied to a wide variety of satellite and other aerospace vehicle development programs.

The reuse of ground station capabilities is most easily demonstrated by returning to Figures 1, 3, and 4. The figures are reproduced here, this time with an emphasis on the extent to which each tool can be applied across programs. Of course, the degree of reusability is strongly dependent upon the commonality among programs. The degree of reusability provided in the figures here assumes that the programs in question are substantially different.

Figure 5 demonstrates the generic nature of the key ground station generation capabilities. The command and telemetry definitions are assumed to be completely different for each program. However, the processes that convert these spreadsheets into C header files for the flight software and database definitions for the ground station can be applied to any program.

Even completely different satellite programs have common elements. Therefore, parts of the command and telemetry screens can be used repeatedly. This is especially true for graphical telemetry displays such as dials and thermometers. These objects can be gathered into libraries that are made available to all screen developers.

Figure 6 shows the program flow originally presented in Figure 3 with information identifying the reuse of work from previous programs. Since each program is assumed to be substantially different, both hardware and software are assumed to be program-specific. For integration and test, the initial test plan development is also program-specific; however, integration and test plans have commoncomponents which can be applied to all programs. The automated test scripts are also custom-designed for each program. However, a significant fraction of the code is responsible for driving the screens used by the test executor. Test execution instructions are unique, but the code necessary to clear screens and indicate test status is completely generic.

Even for completely different programs, all aspects of ground station development can benefit from work performed on previous programs. By using the same ground station software for multiple programs, developers ensure that the data logging formats remain constant. Therefore, the archiving and data analysis tools can be completely or nearly completely reused. This reduces the system design effort to one of determining the command and telemetry system design and interface with the spacecraft.The effort invested in integration and test capabilities also can be largely reused. For example, even for different programs, much of the same instrumentation is likely to be employed. Although the test software may also differ, the code developed to communicate with the ground station can be copied into the new program, thus minimizing the effort required to create this capability. Finally, the screens developed for test execution can be applied to any program.

As with integration and test, on-orbit operational requirements are similar even among widely differing programs. Therefore, a strong degree of reuse is possible. However, the operating characteristics of each individual spacecraft often limit the amount of reuse that can be accomplished in

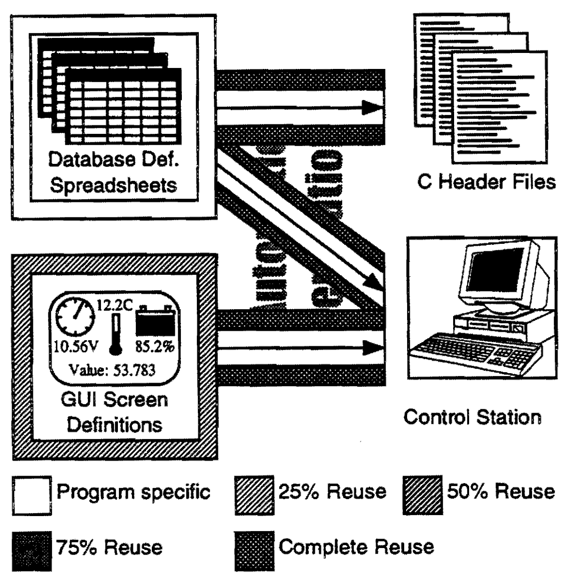

Figure 5. Reusability of Ground Station Development Capabilities.

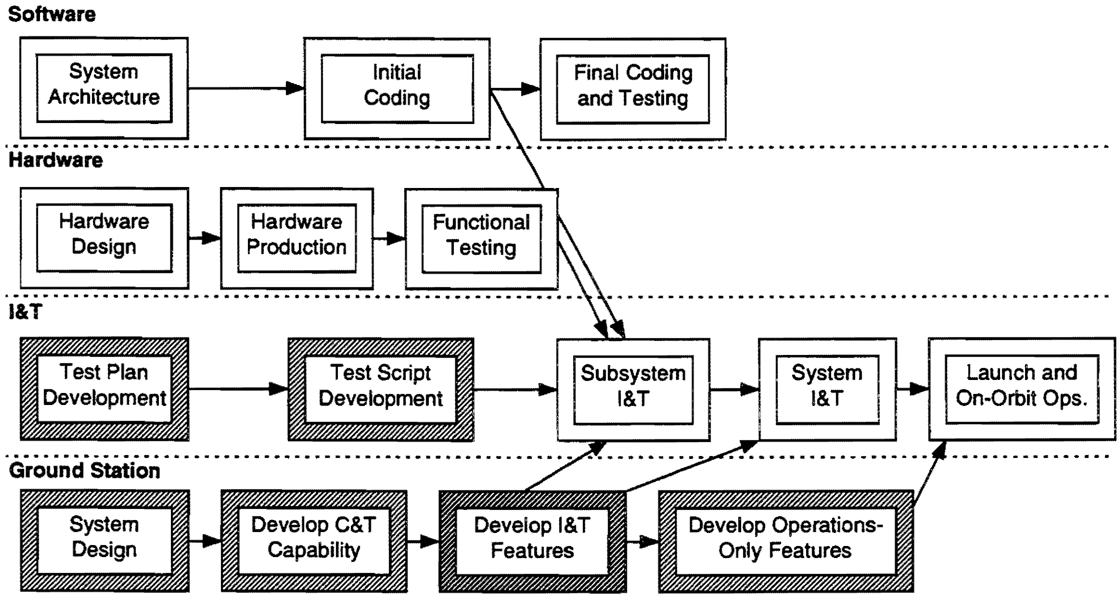

Figure 6. Partial Satellite Program Flow Including Reuse of Effort from Previous Programs.

practice. Nonetheless, some of the most basic capabilities such as pseudo telemetry point computation—for example, calculating power from current and voltage—can be copied from program to program. Furthermore, even if the actual code cannot be copied, existing code serves as a valuable blueprint for developing equivalent capabilities on different programs.

Figure 7 demonstrates the ability of each program to build on the efforts of its predecessors, even when the spacecraft have little in common. The figure shows the tasks used in the ORBCOMM development program. As expected, the launch vehicle simulator and bit error rate test code are program-specific. The data server acts as the interface between the spacecraft and the ground station. If company standards are adopted for commands and telemetry, this tool can approach 100% reusability. Failing this, the data server must be modified to suit the requirements of each program. The world simulation is also program-specific, but it can borrow some components from previous efforts. In particular, environment models for key elements such as gravity, magnetic field, and star locations can be transferred from program to program virtually intact. As mentioned previously, much of the LabView development effort is completely generic. The high level of reuse results from the rapid development ability for new instruments, the use of similar instruments across programs, and the reuse of the ground station interface. Finally, in Figure 7, the combination of all the ground station development efforts already identified are summarized into a single indicator.

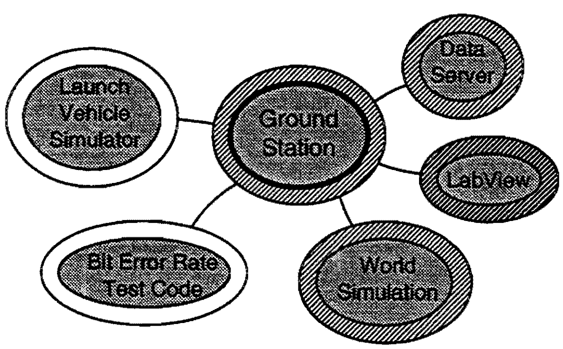

Figure 7. Reusability of Integration and Test Software Components.

Ground Station Selection

The benefits of integrating a ground station early in the program flow can be achieved with a variety of commercially available products. However, the success achieved with this approach is dependent upon having the right tools.

In general, it is preferable to purchase commercial off-the-shelf (COTS) rather than developing custom software. The primary reason is basic economics. When a vendor can sell a product to multiple customers, it spreads its development costs over a wider base, thereby reducing the cost to each purchaser. Second, even if a company has a product which is comparable to or even more advanced than equivalent commercially available tools, it must be able to continue investment in the tool at the same rate as its competitors just to keep pace with the market. Finally, the cost of an internal tool must include maintenance and support costs, which are typically borne by the vendor of commercially available tools.

One common justification for custom-designing a tool is the belief that no product on the market satisfies the specific needs of the program. This can sometimes be a problem of perception or a lack of understanding of the capabilities of commercially available products. However, even if the shortcomings are real, they need not be used to completely rule out a particular commercially-available tool. In the current market, command and telemetry package producers regularly look to their customers for requirements for additional features. Due to the relatively small customer base for these products, individual customers have a surprisingly strong degree of control over the development of command and telemetry packages. If this is not possible, most companies would be more than happy to accept a small development contract to add a particular feature. This is certainly more cost-effective than developing a completely new tool from scratch due to the lack of a few necessary features in existing tools. Therefore, in the long run a company will derive the greatest benefit from relying upon COTS tools where they are available.

For the purposes of this discussion, a ground station is assumed to be comprised of a data server, command and telemetry capability, data recording, data retrieval, and data analysis. The mission analysis/mission planning function is not addressed here, although several COTS products are available for this purpose.

The most important tool is the command and telemetry package. The ORBCOMM program has had tremendous success with the Operations and Science Instrument Support—Command and Control (OASIS-CC) package developed by the Laboratory of Atmospheric and Space Physics (LASP) at the University of Colorado, Boulder. OASIS-CC is an inexpensive, powerful, yet extremely flexible command and telemetry tool. OASIS-CC also fulfills the data recording function. OASIS-CC itself uses the Transportable Applications Environment (TAE+) graphical user interface builder. LASP has been very responsive to OSC input into the design of OASIS-CC. This has included both recommendations for product improvements and OSC-sponsored contracts for more extensive enhancements. The rapid development capability and extensive functionality supported by OASIS-CC have been essential to ORBCOMM's success in realizing the ideals outlined in this paper.

The data server receives telemetry packets over a UDP data socket and converts the messages to the format expected by the command and telemetry package. The reverse process is performed for commands. Although a commercially available solution is desirable, this function i driven almost entirely by the requirements of the command and telemetry interface selected by each specific satellite program, so on the ORBCOMM program, the data server is a custom-developed software tool. If the definition of the data server is extended to include the hardware that receives a radio signal and converts it to data packets (and vice-versa), some COTS solutions are available.

ORBCOMM data archiving is performed using scripts built in the awk scripting language. Awk is a built-in UNIX capability designed for operating on list inputs. The awk scripts take advantage of several UNIX data processing commands. Awk is trivial to learn, but has sufficient capability to meet the relatively limited needs of the archiving function.

Finally, the ORBCOMM program has built its data analysis capabilities using the popular MATLAB software. MATLAB offers the ideal marriage of ease-of-use for novices and extensive features required for sophisticated data analyses. MATLAB's graphical user interface (GUI) is used to create a number of useful data analysis tools. A sample screen from one of these tools is provided in Fig ure 8. All of the functions behind the GUI aredesigned to simplify the writing of MATLAB functions used to perform more complex analyses.

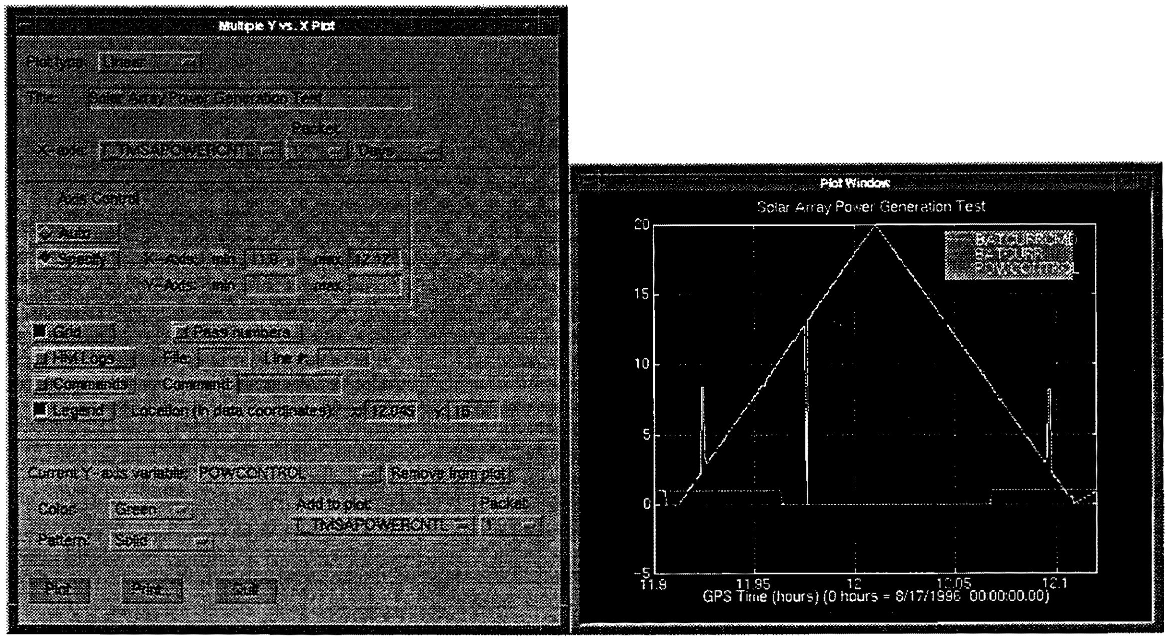

Figure 8. Sample Plot from One of ORBCOMM's Data Analysis Tools.

Summary

The integration of a ground station into the program flow has been shown to produce schedule and cost improvement at each stage of the program life cycle. Rather than simply reiterating the points already discussed, some of the key benefits are presented in the context of the four efficiency improvement factors detailed at the outset.

Parallelization

- Ground station development, automated test development, and software development proceed concurrently

Reuse

- Same command and telemetry tools are used for integration and test and on-orbit operations
- Same data analysis tools used for integration and test and on-orbit operations
- Partial reuse of ground station development across programs
- Nearly complete reuse of data analysis capabilities across programs
- Generic applicability of LabView test hardware control across programs

Process Improvement

- Simple, easy to use mechanisms for specifying command and telemetry messages
- Ground station rapid prototyping
- Automated generation of flight code header files
- Test procedure automation
- Computer control of entire test environment
- "Lights out" on-orbit operation
- Automated data analysis

Task Elimination

- Dedicated ground station to satellite interface tests

As these items demonstrate, including a ground station as an integral part of a satellite development effort produces demonstrable efficiency improvements. Using these methods, the ORBCOMM program has achieved cost and schedule reductions in each of the four categories, which serves as living proof of their viability in the real world.

References

[1]Tandler, John, *Automating the Operations of the ORBCOMM Constellation,* Tenth Annual AAIA/USU Small Satellite Conference, September 1996.