ADVANTAGES OF USING A MODULAR ARCHITECTURE TO EXTENUATE THE EFFECTS OF DISRUPTIVE TECHNOLOGIES

Yason Khaburzaniya  
L-3 Communications

ABSTRACT

Disruptive technologies affect every industry and often indicate the rate of technological advancement. Observing previous technological leaps can help ensure that the next disruptive technology innovation has less of an impact. Many methodologies that were developed to combat the high adoption costs of disruptive technologies in the field of computer engineering can be applied to the satellite and telemetry world. One such methodology is modular architecture in software.

INTRODUCTION

An example of a disruptive technology which affected the satellite telemetry world was the switch from frame-based telemetry to packet-based telemetry. This shift in the philosophy of telemetry delivery had a number of repercussions to ground system technologies and software systems. Software that is not modular entails extensive re-writes in order to handle new technologies. The modular approach to software architecture and hardware design is one way to minimize the effects of technology insertion. Decoupling software systems from hardware systems is extremely important. The urge to reap short term benefits of single purpose, specialized software, may be hard to resist on tight budgets, but the long term costs of modifying software for small technological advancements, let alone technological leaps, may be too high. A smart telemetry system needs to be generic enough to allow the support for both frame and packet based telemetry systems, hopefully making it generic enough to also support the next leap in the telemetry technology.

The modular approach to software architecture and design should have the main goal to decouple software and hardware. Of course, hardware specific drivers must be in place, but these should be a small piece of the software system. This notion has become common in the arena of personal computers. Replacing most components can be done painlessly with the installation of a new driver and sometimes a reboot of the system. The common concepts of plug-and-play and hot-swap technologies can and should be applied to smart telemetry systems.

Platform independence of ground systems is also important in helping to adapt to disruptive technologies. A telemetry system fully dependant on a third party operating system or
hardware component is exposed to the risk of mounting costs, poor maintenance, and product obsolescence. In addition, when a new disruptive technology is introduced, the system designed for a specific OS risks staying technologically backwards until the third party adopts the new technology.

This paper will describe how software development using a modular approach for platform independent telemetry systems can minimize the time and the resources required to accommodate technology advancements and in fact reduce disruptive effects of significant technological leaps.

**MODULAR ARCHITECTURE**

Several reasons inherent to the satellite industry prevent timely absorption of new and robust technologies. The high cost of proving that a component or a system will work in space and will not contribute to a failure of a multi-million dollar mission prevents solutions from being exploited by the industry. Sometimes extraordinary motivation may force a satellite company or agency to abandon a tested space rated or human rated system and replace it with a more advanced, but less mature system. Replacing a single component can prove unjustifiable when dealing with tightly coupled and integrated software due to the high cost of software validation. Major upgrades and redesigns for the NASA Space Shuttle were often triggered by serious malfunctions or tragic accidents. If the Shuttle could be easily upgraded and modified, it would not have to be decommissioned abruptly without having a replacement available.

![Figure 1: Tightly coupled system](image-url)
Figure 1 depicts a system where software is specific to hardware and concepts like automation and commanding maintain dependencies over multiple hardware units. The automation subsystem on Hardware #2 operates closely with the telemetry subsystem, however, some automation tasks are supported only by the automation subsystem specific to Hardware #1. Direct dependencies between the two automation subsystems lead to low cohesion. For such a system changing something that works can be hard to justify. A space program that uses a frame based telemetry system, will not switch to a packet based telemetry system, unless significant financial or operational incentives can be demonstrated or new requirements or capabilities are expected from stakeholders. This type of pragmatic approach over time adds up to usage of outdated technologies and the need for costly overhauls of entire programs.

Telemetry systems that are architected and designed to be modular can help address some of the problems in the satellite industry regarding adoption of disruptive technologies. The cost of replacing an entire ground system and the elevated risk of trying out a new technology must be justified. However, upgrading only a single component may be difficult if the software and the hardware of the ground system were not designed to accommodate for such upgrades. A single component, software or hardware, may be tightly coupled with other components making integration and testing of any upgrades more expensive relative to the potential benefits.

The modular approach to software architecture and design helps to decouple software components allowing for simplified integration and testing of new components. From a validation standpoint replacing a modular component only affects the component and the relevant interfaces. For example, to re-validate the system after plugging in a new module that uses existing interfaces, only the relevant interfaces have to be re-tested. In the tightly coupled system the amount of code re-write and cost of re-validation would be much higher. When implementing the Communication Operations Procedure-1 (COP-1) for L-3’s InControl software, we used Java RMI to plug in the COP-1 module. To re-validate our commanding we had to test the code written for communication with the RMI interfaces. However, no retesting of existing code was warranted because existing code did not change. Also, customers who do not use COP-1 were not impacted.
Figure 2: Modular system

Figure 2 depicts a modular system that supports frame telemetry. Notice the hardware unit is decoupled from the software and can be a single chip, a powerful server, or a distributed over the network computer system. In this example, a telemetry module can be replaced to support CCSDS telemetry. Assuming a dynamic link library (DLL) or a java archive (JAR) encapsulate an entire module, replacing a module can be achieved by overriding a DLL or placing a new JAR file on the classpath. The amount of code re-write is minimal because most changes are contained in a single module. In a large scale system a decrease in the amount of time required for re-validation may influence the decision to adopt a disruptive technology.

In a modular system development of a component can be source independent, meaning it can be open source, developed in-house or by a third party. This added flexibility opens up the field to an increased variety of alternative solutions and increases the ability to customize and upgrade the system (Sanchez). Picking a telemetry solution that supports modular components can prove highly advantageous especially if a single out of the box solution that fully satisfies all the requirements cannot be found.

Operationally responsive systems is a new trend in the industry. In an operationally responsive system, pre-validated modules are put together in a timely fashion to satisfy the demands of a mission. A fully modular software system can easily support this concept.
configuration the system can respond to the demands of a mission by switching in components required for the particular mission.

![Figure 3: Operationally Responsive System]

The system in Figure 3 can be configured to use CCSDS, High-speed, or a Custom telemetry module. Provided that the system supports the three modules, simple configuration changes, such as modifying an XML file or flipping a UI switch, can accommodate the transition from one module to another. Also, no re-validation is required because all the modules have already been validated. With InControl we were able to stretch configuration and modularity to the point where a single ground system can process frame a and packet telemetry from several satellites at the same time. However, for such advanced usage of modular architecture, the robustness of a modular system strongly depends on the quality of the design of the components and the interfaces.

Today many frame based telemetry systems still use hardware decommutators because hardware decom enables superior bandwidth for high speed telemetry. As computer technologies progress, more affordable and reliable software solutions for high bandwidth telemetry have started to compete with hardware solutions. Companies that already own smart telemetry systems with modular architecture will be able to adopt faster. Utilizing a smart modular solution for new projects is beneficial even if no plans to employ a new or a disruptive technology exist. All the benefits of a modular architecture that help reduce the impact of a disruptive technology
Figure 4: Reassignment of Responsibility

Figure 4 displays a telemetry module reassigning responsibility to a more technologically advanced telemetry module located on a completely different set of hardware. Reassignment of responsibility entails re-linking the system and passing the state data from the running telemetry module to the new telemetry module. After this process completes, all the interfaces from the system will work with the new telemetry module and the old module can be shutdown and discarded. To reach this flexibility, InControl makes use of Common Object Requesting Broker Architecture (CORBA) interfaces. CORBA allows TCP/IP communication between modules running on different computers. In contrast to plugging a new module into an existing system, reassigning responsibility is a good option if system requirements of the new module exceed the specification of the current system.

In order to truly be able to take full advantage of modular architecture the software has to be platform independent. The choice of the platform for a telemetry system depends on many factors, however the software that controls the system has to be platform independent, otherwise its flexibility will be pinned to the robustness and the degree of maintenance of the platform. In fact, in the last five years the market share of non-Microsoft operating systems has doubled (“OS Platform Statistics”). No one can guarantee a platform provider’s survival or longevity of the maintenance period for a given platform. A system independent of a single third party product can reduce the risk of becoming obsolete. Also, platform dependence puts artificial barriers to
adoption of disruptive technologies, since it may obstruct better performing and more cost effective solutions. Sometimes a platform dependent solution can provide a performance boost. However, use of modular architecture puts arguments in favor of platform dependent or platform independent solutions in a different perspective. Modular architecture is not about creating a single solution, but more about creating a “platform” for a broad approach to the market (Sanchez). Technologies such as CORBA and the Simple Object Access Protocol (SOAP) enable this “platform” to be OS platform independent. If an OS platform dependent solution proves to be superior to others in the market, it can still be integrated into a platform independent modular architecture. Running on proprietary hardware, platform dependent software can communicate with the rest of the system using platform independent CORBA or SOAP. In addition, generification and standardization become almost natural in platform independent and modular systems, driving costs down in the long run and increasing competition among suppliers. Reduced cost of technology and greater entrepreneurial participation smooth out and speed up the process of transition from one technology to another.

CONCLUSION

Modular platform independent systems can help reduce the amount of time and money it takes to adopt a disruptive technology. In addition, modular architecture supports the trend toward robust, operationally responsive telemetry systems with higher efficiency and improved reliability. In the future, energy storage and alternative fuels could extend the life of satellites. Modular software and platform independence will provide for a greater ability to upgrade both ground and satellite software throughout the duration of missions. Support for IP satellites can be enabled by modifying a single module. It will be possible for software and hardware components to be plugged into satellites on the fly, similar to a USB mouse. In the future many disruptive technologies may simply be modular in nature.

ACKNOWLEDGMENTS

David CheVers contributed to this paper by providing his expertise in satellite telemetry and general aerospace knowledge.

REFERENCES
