MSI TacSat-II Lessons Learned

Terrance Yee Microsat Systems, Inc. 8130 Shaffer Parkway Littleton, CO 80128

Phone: (303) 285-1849, Fax: (303) 285-9880 E-mail: TYee@MicrosatSystems.com

Abstract: The TacSat-II, a.k.a. Roadrunner, formerly Joint Warfighting Space Demonstration 1, mission is being conducted by the Air Force Research Laboratory (AFRL) to demonstrate techniques and methodologies to dramatically shorten the development time required for small satellites. The TacSat-II program is a pathfinder for a 1 year development time, a one week time from call up to on orbit readiness, and 24-hour autonomous on-orbit operations. TacSat-II will show the way ahead for these impressive schedule milestones while fielding a suite of nearly a dozen experiments. Microsat Systems, Inc. is supplying the TacSat-II spacecraft bus based on a design originally qualified for a previous MSI spacecraft. During the course of this program the team has learned a great deal about how to adapt existing designs to new missions as rapidly as possible. This paper discusses some of those key lessons with an emphasis on how they relate to the application of modularity in spacecraft design in particular. These lessons emphasize the success of a "capabilities-driven" approach instead of a "requirements-driven" approach and the extensive use of software modularity and adaptability to meet mission goals.

Introduction

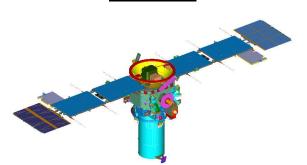


Figure 1 – TacSat-II

TacSat-II, shown above in Figure 1, is the first AFRL mission to directly address the current Responsive Space efforts, which are aimed at drastically lowering the time to place space assets in theater. The primary objective of the TacSat-II (a.k.a. Roadrunner) mission is to demonstrate the rapid development and rapid deployment of

a militarily useful tactical asset. The Roadrunner spacecraft carries a suite of 11 experiments from different organizations and will fly in a 390 km, 40-45 degree inclined orbit. MicroSat Systems, Inc. (MSI) supplied the spacecraft bus including structure, Attitude Determination and Control Subsystem (ADCS), thermal control and some power components. MSI is also supplying bus systems engineering and assembly, integration and test services.

This paper presents some of the key lessons learned by MSI during the production and integration and test of the TacSat-II spacecraft. These lessons have direct applicability to several new missions in responsive space where short schedules are key discriminators.

"Capabilities-Driven"

One of the key enabling concepts for TacSat-II has been the use of a "capabilitiesdriven" approach as opposed "requirements-driven" approach. In this paradigm, AFRL selected payloads and operations concepts within the capability of the spacecraft rather than design the spacecraft to fit a particular mission. This methodology has great applicability to modular spacecraft whose systems will largely be assembled from existing designs. It should also be noted that a primary enabler to this method is having a general purpose, highly capable small satellite bus to begin with. A bus with too-limited capability will have few potential matching payloads. For TacSat-II, the previous MSI program supplied a high-capability solution.

In practice, this approach had to be tempered with the reality of making things actually fit when it came time to finalize ICD's. At this stage, existing payloads met existing spacecraft and it was necessary to craft a workable interface. Manv of the experiments on TacSat-II utilized existing hardware or designs that were completed before TacSat-II to save development time, and could not be easily adapted to fit the bus interfaces as they had been previously designed. When it was not possible to alter a payload interface, the first candidate solution was to adapt the customized harness, whose design finalization was delayed until the last possible date, just 6 weeks prior to the start of I&T.

The chief detriment to this method is that it forced over a dozen last minute changes in the harness that we did not have time to fully review as a whole spacecraft team. The result was that some of these changes were not correct and necessitated modifying the harness after delivery. Generally the

modifications were no more severe than pulling and swapping some pins on a connector, which was easily accomplished. Figure 2 shows some of these modifications being made to the EM harness. A few cases did require running a new length of wire, however. In order to avoid so much rework in the future, more effort should be made to finalize ICD's, including testing EM hardware together, with enough time left to fully review the flight harness design prior to manufacturing. Spacecraft designers should be aware that failure to achieve these two goals greatly hinders the chance of getting the flight build correct on the first try, and could jeopardize their ability to fly on the spacecraft.



Figure 2 - Harness Modifications

If a harness modification could not solve the interface issue, the next place we looked to adapt was the payload interface-specific software, which we called the "payload managers". Since software is one of the easiest things to modify late in the integration process, it became a frequent place to make adjustments for interface mismatches. This meant that we often had to make the software interface work real time after payload arrival. This required a very streamlined, informal procedure for modifying software quickly and getting the new software loaded onto the spacecraft quickly. We were able to optimize this

process to the point of identifying a change, notifying the programmers at Broad Reach Engineering, receiving a new software load and installing and testing it all within about 45 minutes

The key organizational feature that made this response time possible was empowering the line engineers in testing and software to handle each of these steps, thereby minimizing the number of individuals and organizations that had to be involved in getting this type of work done. In order to allow this relationship to develop, key contractors needed enough latitude in how to execute their statements of work without involving a contract negotiation for every minor engineering change. This type of trust is only possible where organizational and personal relationships have been built on the foundation of positive past performance and a commitment to mutual Therefore, maintaining a win-win mentality is essential for bottom line efficiency in responsive space missions. This mission successfully achieved that cooperative team atmosphere. Figure 3 shows that team in action.



Figure 3 - AFRL, J&T & MSI Employees Collaborate on Flatsat Testing

After software, the next level of change implementation was in firmware or loading new VHDL coding into FPGA's used to drive interface transceivers. Due to the type of devices used on Roadrunner, they could only be burned once so this change required new chips to be soldered onto the boards. For EM's this wasn't too costly or risky, but it did take 48 hours to ship the hardware back to Broad Reach Engineering and get it back in the test lab. We only had to make one such change to the flight boards. Repeated soldering and desoldering of these parts stresses the strength of the solder pad adhesion to the circuit boards and should therefore be avoided in flight units where possible.

Finally, if all else is ruled out, changes to bus hardware have to be made. For TacSat-II one change we had to make after finalizing the C&DH hardware was accommodated by a separate board that

converted voltage levels to make the interface work for the Experimental Solar Array IV Box. This board was built by Jackson & Tull, the integration and test contractor on TacSat-II. MSI's analysis team determined where and how the board could be mounted on one of the existing payload boxes. The new board was then put in its own enclosure and adhesively mounted.

One aspect of this methodology that is widely recognized is that these options were much easier to implement earlier in the development than later. Again, every effort should be made to finalize ICD's as early as possible to preserve as many adaptation options as possible. Finalizing the ICD's early also allows the team to make changes prior to releasing the flight designs to fabrication when it is much less expensive and less risky to make modifications. In order to do this, each participant must commit significant effort to finalizing these interface details early in the program and be held accountable for doing so. Neither side of an interface can afford to place all their initial efforts at internal development and neglect the interface to the larger spacecraft. Doing so leads to delays which make it very difficult to accommodate changes.

Software Modularity

On TacSat-II we learned that test software modularity, adaptability, and ease of update became critical parameters for a responsive space program. Due to the number of problems that had to be solved in flight software and the complexity of the payload suite, it became necessary to rely on a much larger, more intelligent suite of test software than previously used at MSI. On TacSat-II the spacecraft and ground test set used Spacecraft Command Language (SCL), developed by Interface Control Systems, to

manage the command and telemetry databases. SCL has a built in scripting capability to allow routine operations to be automated. As payload interfaces matured and changes were made to various mission elements, the SCL integration and test scripts had to be rewritten multiple times to stay up to date and compatible.

To manage the update process, formalized the release procedure for test scripts at the very beginning of I&T. MSI and Jackson & Tull both set up software configuration management databases using "Configuration Verification System" (CVS) servers. MSI used WinCVS clients to access its database, while J&T elected to use a Tortoise CVS client, both of which worked well. For the MSI system, all test scripts and related configuration files were kept in the CVS server and maintained by a single person. This person was not the only one making changes however. The MSI test team on console often made changes on the spot to accommodate features which only became apparent in the actual hardware environment and multiple developers were working in parallel to meet the aggressive testing deadlines. Changes made to scripts were checked back into the system nightly and every week the working directory of the test computer (where tests were run from) was moved to a backup location and replaced with the official CVS version to ensure that no incompatible pieces of test software were accidentally left in the "production" environment. To document changes, each modification was commented with the author, date and reason for change in the code at the location of change. Major updates were also annotated at the beginning of the code in comments and were described in the release notes in CVS.

This system worked very well to track changes and to make sure that script modules staved consistent with each other so that we could maximize the reuse of these modules in other test executives. We set up this system at the beginning and agreed to a set of module standards for controlling the spacecraft settings and configurations when exiting and entering modules. This allowed MSI, Interface Control Systems (a payload software provider) and Jackson & Tull to interchangeably share and co-develop script modules that worked together. Although each organization had different purposes and different styles to their scripts, the common interface standard allowed us to save a lot of development time by coordinating efforts and allowing us to build on the work of the others. The final users, the ground operators of the spacecraft, will also be using scripts built on modules that now have extensive test heritage.

The flight software for the mission has also co-developed been largely with hardware to match the interfaces of the new payloads. This flight software was heavily based on the bus flight software previously developed for another program which allowed MSI to develop compatible test code early in this program. We began to encounter some difficulty keeping up with the flight software growth and modifications once we began integration and test of the payloads, however. Due to their relative newness, these software interfaces were in significant flux early on, requiring several adjustments and adaptations as is typical in a spiral development. These changes in flight software often required new commands and new telemetry mnemonics (variable names) to implement. Due to the fast response of the flight software team to changes, it often became a matter of trial and error to determine if test software was still compatible. Fortunately, compile time error messages would quickly indicate when mnemonics had gone out of synch.

For some of our most frequently used scripts, MSI developed new tools to autogenerate test code to keep up with the demand. We did this by importing the xml databases containing the command and telemetry mnemonics into Excel then using some visual basic code to generate the actual script code to perform our tests. This worked very well for long, routine tests with lots of repetition, but was not worth the effort for other MSI tests that followed less regular, repeated steps.

In addition to this method, Broad Reach developed new techniques in Labview to automatically build & update telemetry display screens based on the most current databases. Display windows would automatically resize themselves to display all the variables in a particular packet or grouping of telemetry and display the textual description of the variable taken straight from the current database. This allowed a lot of information to be displayed in a dense format and largely kept the telemetry system instantly up to date without requiring new human work unless entire new telemetry packets were developed. The disadvantage of this system is that a telemetry item's location on the display screen could change if things were added or subtracted in a particular packet, making it a more difficult for operators to memorize the exact location of critical parameters. Also with this method it was not possible to display critical parameters in ways that made information stand out on a busy screen.

When ICS implemented their RIMS display tool and the operational ground system as an in-house alternative to Labview for data display, these drawbacks in display customization were eliminated because RIMS is very easy for users to modify. While Labview is also generally easy to

modify as well, the particular technique used to automatically keep the displays updated locked us into a single display style. The drawbacks with RIMS are that it has a less rich toolbox of display widgets such as virtual gauges and that only 60 telemetry items could be displayed in a single display window and they didn't pack as densely as This made the the Labview windows. system a little cumbersome when deep in a hundreds debug activity where parameters might be important. other hand, the ability to highlight critical parameters made RIMS easier to use from a spacecraft operator viewpoint since they are more interested in seeing a few things quickly. Since RIMS is used in the mission operations center that TacSat-II will be run from, we switched to RIMS as soon as it was ready to maximize the amount of "testing like we were flying".

With modular systems such as those envisioned by future responsive space planners, the long pole in development could easily be software, so the lessons learned from TacSat-II could have direct benefit in reducing the overall delivery time. Some of the lessons learned on TacSat-II from the RIMS web-based telemetry display and commanding tools have inspired even more far reaching concepts at MSI for spacecraft test & operations that would take full advantage of modern web tools and architecture. At a minimum, responsive space planners should insist on a modular implementation software from their subcontractors

Other Benefits of Scripting

The use of automated scripts for testing on TacSat-II has been both a blessing and curse. On the positive side, we were able to reuse many elements of the automated tests developed for previous MSI projects on this

program and the various organizations working on scripts have cooperated and shared tools and information very well. In fact, two of the software experiments on TacSat-II make extensive use of scripts to accomplish their mission objectives and these scripts are based very closely on the ones developed originally for integration and test. Because Interface Control Systems (ICS, the developers of our scripting language, SCL) is conducting these experiments, the rest of the team including MSI, AFRL and J&T has benefited greatly by having access to their expertise.

The operational scripts, to be used by mission controllers. have also been developed based on the I&T scripts. The I&T team is also using these flight scripts in several tests to ensure that the transition to the flight team goes smoothly and to increase the accuracy of our "test like you fly" approach. One benefit of this approach is that it involves the flight operators early in the development and gives them a chance to suggest improvements that increase the ease of operation of the system and help ensure that the software development on the vehicle is properly targeted to the most useful features.

Each of the groups involved has acted as a source of peer review for each other's work which has resulted in software tools of higher utility and quality than would otherwise be the case. The push to prepare scripts before testing also provided an early impetus to several team members becoming knowledgeable about the test system and the spacecraft capabilities in general well before the first working set of hardware was assembled on a lab table as a "flatsat".

The main benefit of the scripts has been the ability to have non-subject expert testers perform specific scripted activities with accuracy and confidence. This vastly increases the pool of available resources to support testing which thus allows testing to be scheduled with much greater flexibility. The usefulness of this feature cannot be overstated. This enabled us to have a generalist tester run most operations during I&T and if they ran into trouble on a particular test that they couldn't solve, they could get the team back at MSI, Broad Reach, or the payload provider working offline on a solution while at the test site the crew just picked up the next scripted test that was ready and ran that instead. This allowed us to run testing at AFRL with virtually no down time waiting for experts to work out the test kinks. Since we only had one flatsat and one spacecraft to work on, this maximized the testing throughput for our limited resources.

The scripts have also been extremely useful in establishing baseline behavior of the system and components so that as capability is added to the flatsat and new versions of software are released, we can easily, repeatably, perform regression tests that help us identify and isolate problems.

One consequence of performing all the major testing with scripts is that the lead time in preparing tests is much longer due to the necessity of planning each step when coding the script. While this is less of a challenge for experienced operators who are both familiar with the system and the scripting language, many of the contributors to TacSat-II were neither at the beginning of the program. This lead to a large amount of trial and error in debugging the early scripts on the flatsat. As a result, much of the early testing on TacSat-II focused on producing useable test scripts rather than actual verification of requirements on hardware or flight software. This system has the benefit of resulting in knowledgeable spacecraft

operators at the same time as the scripts are produced.

The other difficulty with using scripts is that it can be cumbersome to debug scripts for longer duration, complex tests. For these types of tests, if a bug is found 2 hours into testing, it may take only 5 minutes to fix the bug but sometimes it takes an hour to get back to the point at which the bug occurred and move forward again. The lesson from this experience is that complex flow control such as conditional looping and passing test conditions between subroutines should be avoided to allow better stand alone testing of individual test modules. Construction of tests which run straight through and are broken into several distinct segments allows testing to be reentered for debugging and speeds the overall development process. This also leads to better modularity of the test segments, which can then be reused in other tests with greater ease.

An alternative script generation method which could be tried is to manually run all the commands in sequence, then use the command logs to create the scripts after the fact.

Location of I&T

TacSat-II has a programmatic objective of building AFRL corporate knowledge to be able to pass lessons on to all of industry. As a by-product of this objective, there was knowledge transfer about considerable small, fast satellite missions to the junior grade officers and other support personnel at AFRL. For AFRL to gain the most knowledge to pass along to industry, it was necessary for the program must have as much activity take place at AFRL as possible. This meant that from component checkout to assembly, integration and test of the spacecraft bus occurred at the Aerospace

Engineering Facility and building 277 at Kirtland AFB. This had the secondary benefit of allowing a large number of other AF personnel to witness the work and get a real sense and feel for how the new paradigm actually worked. This has certainly helped fuel the rapid increase in visibility of these techniques and the greater acceptance of them within DoD.

As successful as this decision has been in achieving these goals, it did not come without a unique set of challenges. First, access to the test area was not available to MSI outside of normal work hours all the time partly due to the need to be escorted by on-base personnel who could not always be made available at the odd hours testers needed to keep. Similarly, base security rules severely limit internet access, even wireless internet access, to only cleared personnel stationed on base full time. Fortunately, permanent base personnel were not constrained by these restrictions.

Second, having to run some operations through other people instead of doing them ourselves is inherently slower, but necessary in order to teach properly. Sometimes this was also necessary due to the fact that MSI personnel were not trained on all the different equipment at AFRL.

Lastly, there were many operations that required the design experts to be on-site and required well over a man-year of combined travel and extended time away from families and homes. Obviously, MSI prefers to minimize travel to minimize the hardships to its employees.

These challenges were met by the combined TacSat-II team of Jackson & Tull, Air Force personnel and MSI through a combination of dedication, hard work, flexibility and resourcefulness. This use of scripting as

mentioned above was one of the key features that allowed the team to maximize its efficiency and keep the program on track. The lesson to remember is that unless the mission must integrate at a customer's site for knowledge purposes, MSI finds it much more efficient and straightforward to assemble and test our bus product at our facility where we have full access to our local subcontractor support teams and the other MSI support systems and personnel during I&T.

Other Lessons

Keeping the appropriate level of documentation and checks and balances on the program was crucial to keeping the fast pace of our program without getting out of control. For I&T the two critical documents were a daily Test Conductor's (TC) log and a database of Problem/Failure Reports (PFR's). The TC log was a one or two page summary of what had occurred that day and a listing of any new techniques learned or other useful tidbits of knowledge. Kept as a single word document that was appended daily, it included all the knowledge gained on previous MSI programs as well as TacSat-II and was used several times a day to search for information and special methods relating to all aspects of the The PFR's served a similar satellite. purpose in making sure problems got corrected and not forgotten and that lessons were properly learned during the program. For checks and balances we always had two engineers working on the spacecraft or flatsat at any time, always using a second set of eves to check touch labor. At the same time most of our touch labor activities did not need anything more than engineering approval from the MSI or J&T cognizant engineer.

With the large number of different organizations involved in TacSat-II, full participation by each party in our weekly program meetings became very important. If one organizational representative was missing, or delinquent in a deliverable to the group it was quite possible to have that one schedule hit ripple into each of the individual schedules of the 20 or so organizations. Therefore, for missions requiring many cooperating entities, it becomes even more critical that each does its utmost to meet the deliverables to the others and all are active participants in group meetings.

Conclusions

TacSat-II has developed a wealth of new techniques and methods to speed the performance development of high microsatellites. By performing I&T at AFRL we have provided the opportunity for these lessons to be communicated to a large community at the AFRL, in the Air Force in general, and among the civilian partners and participants in the mission. These lessons, such as how to use a capabilities-driven approach and software modularity, form a set of building blocks even ambitious for more microsatellite missions for the Air Force to Missions such as TacSat-II are enabling a broader shift in thinking within the industry about what the limits of responsiveness can be in this field. By applying the lessons of TacSat-II and fully developing the ideas inspired by this and other responsive space missions, industry is preparing to implement the dramatic jump to space missions that respond to changing tactical demands in a matter of days. In addition, TacSat-II and the other missions in the series are creating many individual advances that can be applied to medium and even large spacecraft.