ABSTRACT: The paper follows the entire product evolution life cycle, illustrating how small, low cost space engineering and design methods have been translated to the design of a subsystem. This description includes subsystem specifications, technologies, materials and layout, operating modes, development and test methods – including prototyping and achieving flight ready status – and even extends to the marketing approach now being instituted. AeroAstro has successfully executed this low-cost approach for subsystems ranging from optical sensors to radios and actuators with consistent results. We show in the paper how the lessons learned in 20+ years of microspacecraft experience can and should be translated into other design and engineering projects to achieve breakthrough products enabling new missions for lower cost while increasing performance and reliability of proven microspace applications.

As microspace has evolved beyond LEO to GEO, Lunar, and even interplanetary applications, the need for a star tracker compatible with satellite power, mass, volume, and cost constraints has become both compelling and obvious. Or so we thought, writing proposals to fund its development in the early ’90s. The response from the microspace customer community was discouraging – repeating criticisms of the microspace movement in its earliest days. Why pay money to develop a star tracker with lower angular resolution, lower light gathering capability and lower quality components when the field had evolved far beyond our proposed specs? Of course in consumer product engineering, we know that the largest markets – where the most people are served and hence the highest utility is provided – are not at the extremes of performance, but at the low end of performance. The world’s best selling airliner is the 737. The biggest revenue producers for automobile manufacturers are not Corvettes and Maseratis, but rather the Toyota Camry and before that, the Ford Taurus.

We know that the engineering of a PC priced for the mass market is completely different than a custom device optimized around extremely computation intensive applications like complex signal processing or managing a node on the national Internet backbone.

Requirements formulation is the critical step in any microspace design. Years of pushing performance specs has made emphasis on this step particularly important, since our experience biases us toward inappropriate optimizations. Rather than aiming for the “best” performance, we should seek the poorest specifications – for angular resolution, bandwidth, light sensitivity, radiation hardness, reliability and lifetime – that can successfully accomplish the job. Microsatellite missions offer a trade off – in this case shorter lifetime and poorer pointing performance – in exchange for advantages that simpler missions and systems offer: lower cost coupled with higher reliability, the ability to fly multiple spacecraft, and spacecraft tailored for specific boutique missions.

By minimizing requirements to barely achieve the mission, we create a virtuous circle. A simpler design is intrinsically more reliable, hence we can build highly reliable devices without resorting to the most expensive, heritage components. Use of more modern components further reduces parts count, increasing simplicity and reliability, while further lowering cost, size, mass and power consumption.

But securing funding in a performance-focused industry, for a thrift-optimized development, is at best difficult and time consuming. We employed the usual methods – cultivation of a few believers (we would label them visionaries) and stitching together small SBIR, STTR and BAA contracts each addressing a small part of the system. For example, we combined two non-overlapping SBIRs from different agencies to complete a prototype Miniature Star Tracker (MST). One SBIR focused on the development of the imager...
and its basic functionality while under another SBIR we developed the advanced processing algorithms implemented in the MST such as Lost-In-Space quaternion generation. Figure 1 shows the MST Engineering Development Unit built under these contracts, primarily used to test and verify operation.

Also, to minimize development cost, we teamed with organizations who already possess solutions to part of the puzzle. For example, we teamed under STTR support with a group at MIT who had developed centroiding algorithms capable of sub-pixel accuracy. Management worked to steer engineers away from their inclination to build a new project from a blank sheet. Rather we led with the principle of “standing on the shoulders of giants”, emphasizing design by emulation, which begins with background research to determine if there are processes and algorithms already existing in the literature that serve our purpose or can adapted. These often served as starting points very far down the development path. Leveraging work performed by others, often done at taxpayer’s expense, not only lowers our development cost and avoids relearning many lessons already hard-learned, but supports the sponsor by demonstrating the value of previous investments made by the sponsor, and helps in sales since many potential customers are positively inclined to buy a product incorporating elements the customer organization developed or funded.

Restraint from the lure of the blank sheet of paper requires more than just management encouragement. Use of existing solutions is often seen as a compromise – for example utilizing a (software) solution or (hardware) component optimized around some other application. A culture of improvisation and ingenuity, similar to the skills many hams have, of building useful items out of parts you have on hand or can readily scrounge, must be nurtured in a setting where they usually are not – development of componentry for space application. But looking at low-cost designs in other industries belies the aerospace bias toward top-down design. Multiple automobile models tweak their designs to share common components. Computers spanning a range of price and performance are designed around readily available displays, storage devices and processors. In our case, we proposed a miniature star tracker built from a COTS CMOS array designed for consumer cameras and cell phones running image analysis software already developed at MIT for use on scientific telescopes.

If low-cost systems design were simply a matter of restricting ourselves to cheap components, minimal requirements and existing components, it might still be a valuable discipline, but not as interesting nor productive. But often to achieve the goals of low cost and simplicity, innovative engineering is unavoidable. To produce the DVD player you can buy at Wal-Mart for $49 required development of costly custom ASICs and investment in the engineering of innovative mechanisms for the read/write heads that are both cheap to produce and rugged in the field. Undoubtedly, the product’s design leveraged the capabilities of low cost manufacturers combined with existing components and packaging.

In the case of the MST (mockup shown in Figure 2), while microsatellite missions do not require as high pointing accuracy, the spacecraft cannot afford the cost, power, space and complexity of additional components: gyros, for measurement of higher angular rates, up to 2 RPM, and additional instruments to bootstrap the attitude (the so-called “lost in space” capability). Thus while we might minimize the number of pixels of the sensor to simplify computation at the expense of angular resolution, and thus significantly shrink the optics aperture and complexity, we do require
algorithms and signal processing that can sense attitude at relatively high slew rates and can bootstrap with no prior attitude estimation.

Conventional spacecraft programs proceed linearly from requirements through concept, preliminary and critical design, then parts procurement, assembly, integration and test. In microspace, however, we begin prototyping immediately to avoid investing too heavily in designs that don’t ultimately work out, and leverage the simplicity of the design to get hardware built early to avoid a prolonged process of over-design.

Under Phase I STTR support, we began construction of an Engineering Development Unit (EDU). Using readily available COTS components, we developed software that could accept limitations including the pixel-to-pixel variability associated with lower cost CMOS imagers, as shown in Figure 3. Software was developed in a simple simulation environment ahead of the development of host hardware, using readily available, open source (free) development tools. To leverage software elements already developed for other applications, we strictly enforced a modular software architecture. We invested relatively more effort in finding existing software and fitting our architecture to it, and relatively less resource in the relatively risky process of software development. Many of the mathematical routines, such as eigenvector/value calculations, were quickly implemented from open source or other zero-cost sources to bring the EDU to an operational state as quickly as possible. Over time, these routines are being tested to determine if they are suitable for use in the final product, or if they need to be replaced to achieve higher performance or robustness. Figure 4 shows an approximate breakdown of the utilization of existing hardware and software in the development of the MST EDU versus developing new.

The importance of generating hardware quickly in the program to push development cannot be overemphasized. PowerPoint engineering is a necessary part of the business to sell the product concept, but once the concept is funded, the goal should be to immediately produce something that can be touched and felt. Achieving basic functionality – and then twiddling knobs and adjusting parameters – is highly motivating and educational for the engineering team and management, both on the developer and the client side of the project. These early successes motivate participants and stabilize the resolve of funding sources. Hollywood did not make a hit movie of “those magnificent men in their simulators.” No, it was their “flying machines” – the actual hardware – that elicits praise, respect and interest. In our case, the hardware itself drives the program forward, because it makes increasingly obvious what functionality needs to be enabled and (perhaps just as critically) at what stage of the program. Simulated data are better than nothing, but actual real-world data are more reliable, more instructive and more exciting.
The majority of microspace missions that do not function on orbit do not fail after launch. They fail to ever be launched because they do not obtain, and maintain, the necessary funding. Early fabrication is the life insurance payment that simultaneously advances product development. By showing early hardware to our customer base, still dubious of the value of a lower performance unit, confidence is gained that a useful product will really result despite tight development cost constraints. Furthermore, in hardware and software, we can demonstrate highly innovative approaches which appeal to the aerospace bias toward innovation. Maybe most importantly, armed with real, albeit prototype, hardware, we can elicit and demonstrate interest from other government and industrial customers, which also tends to fortify the sponsors’ resolve. Simultaneously, the existence of working prototype modules reduces the residual risk, eliminating another obstacle to continued financial and technical support.

Another classic element of microspace development incorporated into the MST program is our approach to testing. Like voting, we advocate testing early and often. But we do not advocate spending precious money on high fidelity test facilities. We use whatever environments exist to test the device. In the case of the star tracker, a backyard in rural Virginia on a clear night provides a star field suitable for algorithmic and optical sensor testing. While we cannot cheaply provide a high fidelity simulation of the space environment, the back yard test provides real world calibration data on the sensor’s performance which we can compare to our design-based expectations. Verification that our model of the hardware accurately predicts back yard performance, allows more confident projection of in-space performance. If the MST algorithms work under the constraints of atmosphere and the occasional airplane drifting through its field of view, it will work under the less cluttered and less optically turbulent environment of space. The margins for algorithmic performance are automatically built-in with the testing conditions.

Ultimately, it is nearly impossible to sell innovative products for space application without in-space heritage. Here too, innovation can lower the cost barriers to product success. Working with partners at Up Aerospace recently, a suborbital payload slot became available to flight test the unit. While not a full orbital test, the suborbital flight provides a launch environment at least as challenging as an orbital launch,
as well as a few minutes’ exposure to the space environment. As of this writing this critical first flight is scheduled for mid-August, 2006, and will be reported at this paper’s presentation.

Currently, the MST unit is baselined on a variety of space missions, and in addition is eliciting interest from large manned missions. We are also exploring several terrestrial applications for the MST hardware. Customers have requested us to provide several enhancements to the basic design.

The importance of securing a supportive market cannot be overestimated for any space development – spacecraft, human space flight, or even a small component. Ultimately, society must make a conscious choice to pursue space, as at least in the short term, space development isn’t required to put food on our tables. Funding agencies want to support programs that satisfy intermediate markets (spacecraft developers) and the ultimate customers – the US DoD, space science community, and the American population. We invested our own resources not just in augmenting what could be accomplished with external STTR and team member support, but also in generating buzz. We wrote papers for conferences and journals, set the EDUs on card tables at conferences, and continue to visit current and potential customers to elicit their feedback on the design, performance and features.

Microspace exists today because of the sustained efforts of its pioneers, who insisted on its utility in a world dominated by giants, and who developed technologies and methods tailored to low-cost, low-complexity space systems development. Our success with MST is based largely on the lessons learned in decades of microspace work. Our goal is to further enhance the reach of microspace through successful introduction of a range of components – including sensors, radios and power systems, matched in their design and capability, to the missions microspace addresses today and in our future.