AIAA <i>Infotech@Aerospace</i> 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

AIAA 2007-2911

AIAA Infotech@Aerospace 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

A Guidance, Navigation and Control (GN&C) Implementation of Plug-and-Play for Responsive Spacecraft

Submitted by:

Paul Graven
Microcosm, Inc.

L. Jane Hansen
HRP Systems, Inc.

It is becoming more important to the space community that spacecraft be developed in direct support of responsive missions, which implies that spacecraft need to be "built to inventory", constructed from off-the-shelf components based on varying mission needs, and/or integrated rapidly with payloads as they are identified and available. Microcosm, in conjunction with HRP Systems, under several SBIRs from AFRL/VSSE, has developed a rapid prototype implementation of Plug-and-Play (PnP) components and generic core algorithms that are focused on a guidance, navigation, and control (GN&C) application for spacecraft scheduled for demonstration on the AFRL PnPSat. This paper will address architectural issues that drive a PnP implementation of the GN&C function, with an emphasis on identifying and understanding all of the pros and cons associated with this implementation path. Additionally, the paper will examine the Microcosm/HRP Systems approach to GN&C design, including software partitioning, algorithm development, and performance testing.

I. Plug and Play (PnP) Introduction

Through the transformational communications activities within the government, the application of today's internet technologies are being exploited to create a robust, network-centric conglomerate of space assets that will communicate in much the same way that personal computers communicate through the internet backbone. At the level of a single spacecraft (and beyond), there is the implication that spacecraft will need to embrace the PC-based concept of Plug-and-Play (PnP), where the user plugs a mouse into a PC resident USB socket and invokes the operating system to find the correct driver, configures the system parameters, and then seamlessly allows the user to begin operations. Small satellites provide an excellent, near-term test platform for proving new spacecraft technologies. They can also provide a responsive capability for missions that necessitate quick launch, quick operational capability, with an attendant, very short development schedule. An attendant benefit is a lower cost mission. Creating a truly modular, PnP spacecraft capability will allow development cycle times to shrink to the order of a few months instead of the usual few years, which will enable rapid testing of new space hardware, as well as creation of highly responsive space systems.

The application of PnP to the GN&C discipline is of particular interest because of the flight critical nature of this subsystem function within the spacecraft. The GN&C subsystem requires real-time execution, as well as timely data delivery with known, relatively small, latency. Conversely, a PnP implementation increases scalability, availability, and fault tolerance of the GN&C application, which ultimately leads to a more robust subsystem functional element. The goal is to balance the pros and cons, while assuring that the resulting GN&C implementation provides the performance attributes typically required by more traditional, much longer schedule and much more costly, subsystem development efforts.

As the term operationally responsive space (ORS) has become more accepted in the spacecraft development milieu, the application of today's internet technologies are being exploited to create a robust, network centric conglomerate of space assets that will communicate in much the same way that personal computers communicate through the internet backbone. At the level of a single spacecraft, the implication is that spacecraft will need to embrace the PC-based concept of "PnP", where the user plugs a mouse into a PC resident USB socket and invokes the operating system to find the correct driver, configures the system parameters, and then seamlessly allows the user to begin operations. Microcosm's work on this application of a self-configuring network for PnP applications is focused on achieving this vision of spacecraft built to inventory using attributes of the internet protocol to achieve rapid configuration and re-configurable communications. Of primary concern and the focus of the work discussed in this final report is the implementation of GN&C algorithms to take advantage of the PnP nature of sensor and actuator components.

The concept of a PnP implementation is rooted in several other, more traditional, concepts, which have been embraced and embellished by the commercial world to achieve the PnP attributes of today's personal computers. Standardization, open architecture, commonality, and reuse are all elements of accepted commercial standards that are now being investigated by spacecraft manufacturers. The primary reason that these concepts are being investigated, and in many cases accepted, is to achieve a more responsive capability for space-based resources. Standardization implies the use of "common" elements across manufacturers, across applications, and across missions. Standardization can apply to hardware, firmware, and software, as well as architectures and interfaces. It is through the use of such standards that the PnP paradigm can be realized for space assets and resources.

Hardware standards have progressed radically over the past 40 years, and the advent of the field programmable gate array (FPGA) technology has accelerated the progress. Both hardcores (hard wired) and softcores (programmable) are now available for implementation in standard FPGA formats to perform functions that would traditionally be thought of as specifically analog, digital, and software implementations. The FPGA also affords the opportunity to create environmentally tolerant components using these commercial off-the-shelf (COTS) functional cores. Further, most computer instruction set architectures (ISAs) now have cross-compilers for standard ANSI C (and often even C++) thus encouraging standardization in software. This hardware, firmware, and software standardization has come a long way and is now well established, even in the development of embedded flight software. It is the standardization of processing architectures and the associated interfaces that will make spacecraft components truly common and thus reusable, a goal that has eluded the embedded flight software development community for decades.

II. PnP Applied to Guidance Navigation and Control (GN&C)

From the perspective of a guidance, navigation, and control (GN&C) application, there are three general elements that make up the architecture: the sensor(s), the processing, and the actuator(s). The general philosophy of development is to abstract the core software components from the specific sensor and actuator suite(s) by establishing standard data elements and descriptors. The overall goal is to remove traditional subsystem component boundaries, while defining interfaces in terms that remove vehicle or subsystem component information (i.e. serial number, mass properties) from the implementation. This implementation approach, while antithetical to traditional GN&C development, will include significant innovation, by creating a processing architecture that is data centric. A data centric architecture implies that the specific devices and/or subsystems are no longer the system drivers, but rather the new driver is selecting the correct vehicle configuration to meet the specific mission needs.

As shown in Figure 1, if the central element, the processing, can be abstracted from the details of the source (sensor) or destination (actuator) of the associated data, that element can become common to multiple mission applications. While this common processing element may not be specifically optimized for each individual mission, it could be created in general categories that are "good enough" to meet 90% of that categories mission needs. Categories might include high accuracy pointing/mapping for imaging, versus, less accurate knowledge and control required by some communication devices or low resolution, large area cameras. To further refine good enough, the processing could also be made generic and adaptive, such that limited information about each data element (sensor/actuator) allows the reusable, common processing core to tailor filtering styles and/or gains. Finally, on-board simulations that can support the adaptive nature of the processing will be needed to allow for on-orbit calibration and tailoring.

The development of a generic, reusable ADCS processing core implies that there is no implementation optimization for a specific mission objective. While this is true, the risks associated with the approach can be mitigated through the optimizing the quality of the data input into the core processing algorithms in conjunction with providing multiple data combinations to establish the "best" knowledge estimate of vehicle state. Additionally, the overall quality of the vehicle state estimation can be enhanced through the use of upgraded or higher accuracy sensors. With that in mind, the development of a mechanism or framework to include all available sensed information at any given time, including non-homogeneous data, data that is output at different or varying data rates, and intermittent data will enhance the state estimation process. By using sensed or measure data, in conjunction with synthesized or derived data, and including analytical data measurement models or simulation to fill in the gaps, an optimal solution can be achieved.

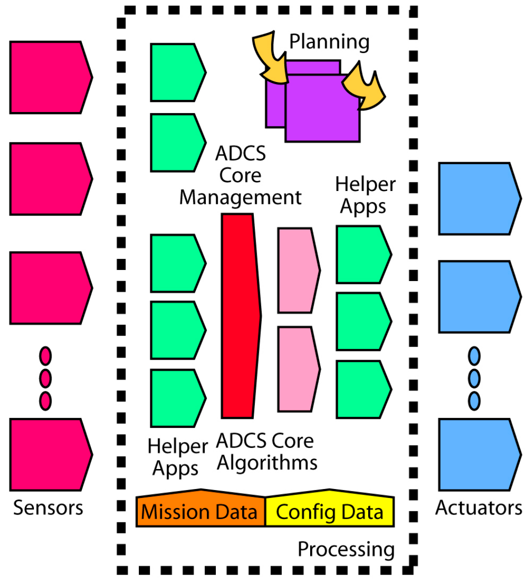

Figure 1: Abstraction of the core processing from the sensor/actuators can create a flexible, scalable system that is not dependent on the sensor suite, the actuator suite, or the specific vehicle characteristics.

The use of an adaptive, generic filter that can optimize the determination of the current vehicle state (position, velocity, attitude, attitude rate, etc.) given a varying measurement set from different sensor suites, provides a mechanism to assure that the core processing can stay common under differing circumstances. This framework would allow for a broad array of traditional NAV and ADCS sensors, as well as the integration of "synthetic sensor inputs" which could come from specialized payloads, communication devices, or ground interaction. The process of developing, implementing, testing, and receiving acceptance of these new algorithm technologies will require a long term view and may require more effort to implement initially than traditional approaches, but with optimum payoff in the long term. Adaptive, generic position and attitude determination and control implementation challenges, such as an up-front investment that must be made in design and development of software along with the detailed understanding of potential sensing devices and their required interface specifications are needed to fully realize the PnP benefits. However, once this investment is made, and the algorithmic advancements are embraced, the adaptive, generic control framework could become the new baseline approach for next generation spacecraft navigation and attitude control systems. The implications are that this new framework represents a disruptive technology (Clayton Christiansen, Innovators Dilemma ref.), meaning that it provides potentially lower performance (good enough) at lower cost, that through continued improvements will eventually replace traditional technologies, specifically with respect to back-up or fault tolerant systems.

As illustrated in Figure 2, the adaptive, generic filter accommodates potentially varying quantity and quality of inputs, while maintaining "acceptable levels" of performance throughout. These filtering and propagation techniques can provide good-enough steady state performance without requiring formal mode switching. Thus, the filter will operate at system start-up when the data is not calibrated or refined, and the filter can continue to function even is data is temporarily unavailable or of low quality. Generally, this generic, adaptive filter allows for a broad array of traditional NAV and ADCS sensor inputs, as well as the integration of "synthetic sensor inputs" which could come from specialized payloads, communication devices, or ground interaction to achieve an estimate of the space vehicle current state.

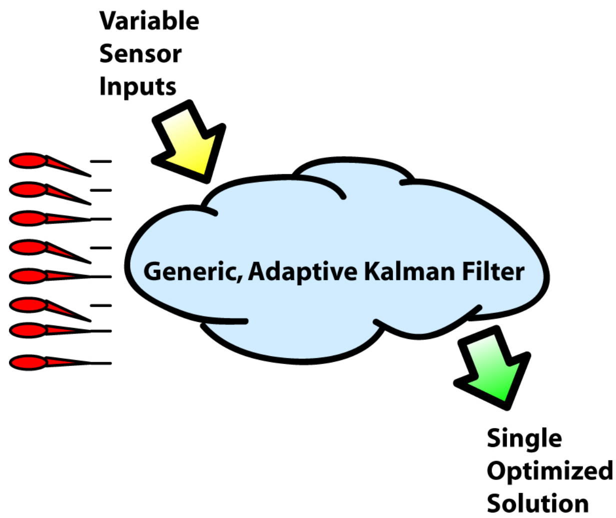

Figure 2: Generic Adaptive Filter allows for a broad array of inputs to provide a continuous, robust determination of the vehicles current state.

Taking this concept further, if the actual control problem is reduced to a generic question of what is the current estimated state of the vehicle (time t), and what is the desired state for the vehicle at time t+1, a generic set of algorithms can be developed to solve multiple problems. As shown in Figure 3, whether the control mode is sun pointing, ground track (maintaining lock on a nadir point as the space vehicle moves), or ground stare, the algorithmic implementation can be the same. The sensors used to measure the current state may vary with the objective (formally termed mode) but the evaluation of how different the current state is from the desired state is a static function. The mechanism to affect change on the vehicle from the current state to the desired state will be impacted by the actuator(s) available, however, the determination of the necessary rates, translated to torques through the inclusion of vehicle mass properties, will be independent of the mechanism used to impart the torque.

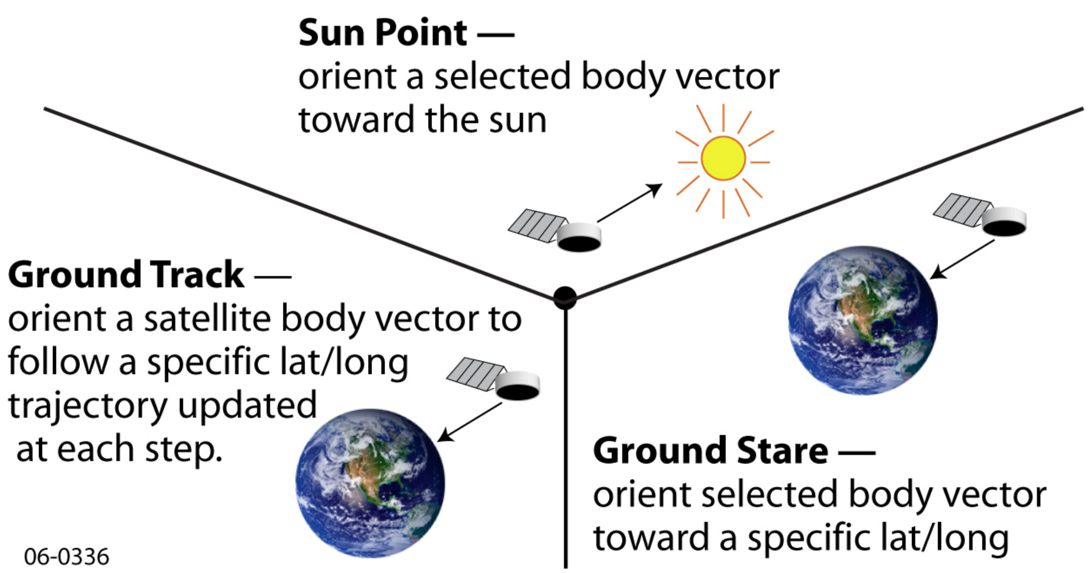

Figure 3: Inputs, Processing and Outputs are the SAME regardless of what is pointing where; there is one control law designed to reduce the error between sensed and desired state

Within this paradigm of one control law, there still may be varying modes of operation that will be performed with respect to varying objects. The objects or pointing directions are defined to include planetary objects (Sun, Moon, etc.), Nadir defined as perpendicular to the earth surface, zenith defined as the opposite of nadir.

Additionally, the ability to point to a vector (position) or target (state vector including rates) can be accommodated. These modes will include the following:

- SLEW — Slew to a vector (time, position, velocity)
- TRACK — Maintain pointing to a vector (time, position, velocity)
- RATE — set body rates to a vector (de-tumble)
- STANDBY — transitional or cruise (minimum power consumption)
- MOMENTUM_DUMP — reduce stored momentum

III. PnP Software Architecture for a GN&C System

The software architecture or software partitioning is a critical element of the design process that will be address to meet the goals of commonality and reuse. With an emphasis on GN&C, the goal of functional partitioning hinges on the ability to identify atomic data elements. These are pieces of data with a direct tie to physics, rather than subsystems or components. For example, spacecraft motion, rates, expressed in body coordinates. These would be considered atomic level data rather than velocities measured in sensor coordinates via an inertial measurement unit (IMU). Another example is a vector (line of site) and clock angle to the sun, in spacecraft coordinates, rather than an intensity measurement of light on a sun sensor in sensor coordinates. In terms of outputs from core GN&C processing, desired rate, imparted on the spacecraft in spacecraft coordinates rather than a torque (which includes vehicle mass properties) or worse, on/off time for a thruster. A listing of the basic atomic level data elements[1] in a low earth orbiting (LEO) space vehicle are as follows:

<u>Sensors</u>

1. Time — time stamp
2. Rotation measurements — rotation angles (3 components), rotation rates (3), rotation accelerations (3)
3. Translation measurements — translation position (3), translation rates (3), translation accelerations (3)
4. Third body angles — Earth angle (2 components), Earth angle rate (2), Sun angle (2), Moon angle (2), and star angle(s) (2 components for each star in the field-of-view).

<u>Actuators</u>

1. Requested thruster force (3 components)
2. Requested thruster torque (3)
3. Wheel momentum and torque (2 per wheel)
4. Magnetic torquer dipole (3 per torquer)

Similar to abstracting data from the source, as with these atomic data elements for inputs and output to the core, GN&C processing can also be abstracted. In the abstraction of the core processing, low level data elements can be created to express quality and type of information provided to and required by a sensor or actuator, respectively. This measure of atomic data quality would be used to determine the source and destination of processing inputs and outputs, as well as facilitating a mechanism to refine adaptive algorithms. This concept of allowing for varying levels of fidelity and accuracy based on the type and quality of data that is available allows the core processing to be scalable and flexible to meet varying mission needs. Likewise, the ability to control the vehicle is based on the control authority that is provided by the specific actuator or actuator suite that is available. At the top level, Figure 4 illustrates the idea that the quality of attitude knowledge and ultimately vehicle control is directly related to the quality of data that is available to the system.

[1] Originally presented as part of SBIR contract F29601-02-C-0240: Application of Self-Configuring Network for Plug-and-Play Device Control with AFRL, then again as part of AAS-06-034Achieving Responsive Space: The Viability of Plug-and-Play in Spacecraft Development

AIAA Infotech@Aerospace 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

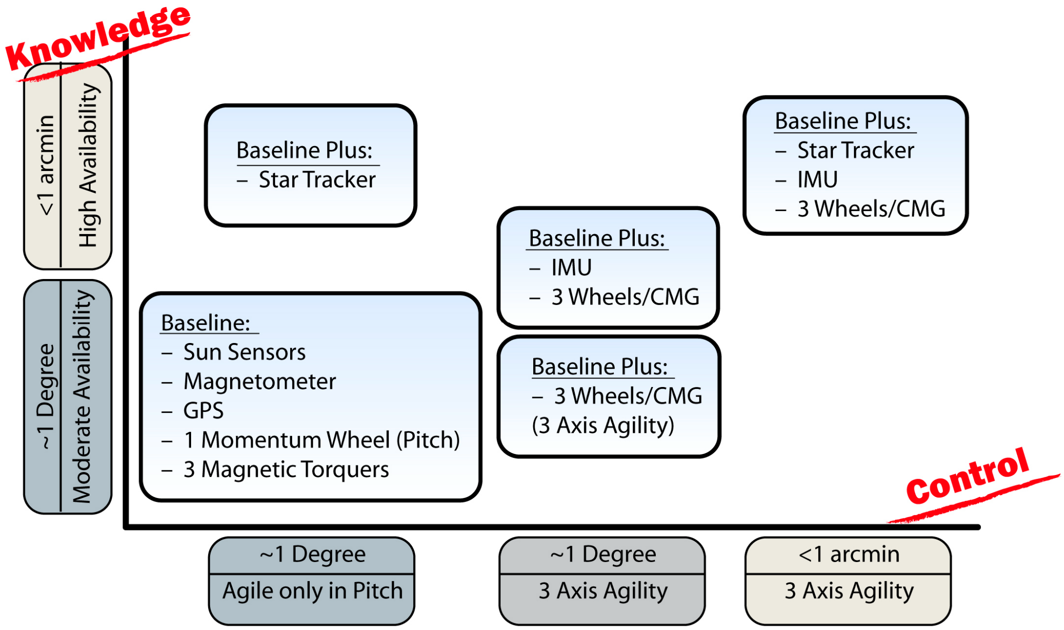

Figure 4: Spacecraft Attitude Knowledge and Control are directly related to the sensor/actuator suite that is available.

Utility functions such as mathematics and environmental models follow a similar development approach. These multi-use software components should be robust enough to handle varying mission requirements and orbital conditions, allowing for testing once and multiple usage. Embracing the concept of multi-use and reuse functional components, leads to an implementation approach that includes the notion of helper application software modules (helper apps). These helper apps are developed to provide a single, low-level processing module. Examples of the various types of helper apps that might be created are:

- Kalman Filter for state estimation
  - Synergy if over-determined or multiple data sources
  - Mathematical noise reduction
  - Synthesis of estimates for states not directly measured
- Translate new or different device to standard data elements – such as
  - GPS pseudo-range versus position/velocity outputs
  - X-ray pulsar measurements for position determination
- Sensor Data
  - Vector to Sun
  - Vehicle rotation rates
- Keep-Out Zones for Payloads or other instruments
- Coordinate Frame Transformation(s)
- Time Conversion(s)
- Propagation
  - Orbit
  - Attitude
- Disturbance torques, drag models, gravity models, measurement models, etc.
- Earth, Sun, Moon, Solar System Ephemeris

As shown in Figure 5, the building up of the generic helper apps, the sensor/actuator translation helper apps and the abstracted core processing components leads to the creation of a layered software architecture, much like an onion or garlic pod. The core components are not dependent on the sensor/actuator suite, and the software utilities are robust enough to handle varying conditions. The ultimate mission configuration and detailed sensor/actuator component description is maintained as part of the external layer of the architecture, in the vehicle configuration data.

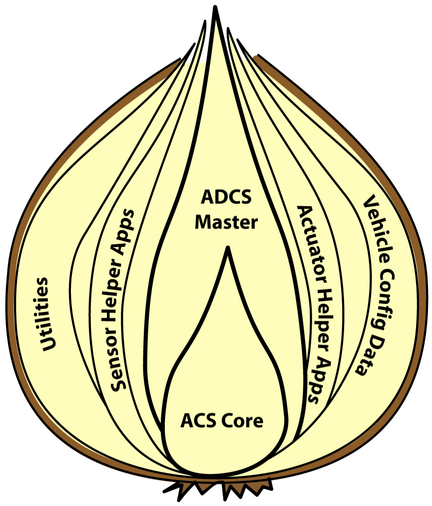

Figure 5: The ACS Core is abstracted from the sensor/actuator suite while the layers of vehicle specific information and support algorithms build up the size and complexity of the overall system.

Typically, within this PnP paradigm, the type, quality, and "unpacking" instructions for each piece of data are held within a run-time parsable ICD that has been labeled as an extended transducer electronics data sheet (xTEDS). The format of the xTEDS is an industry-standard XML notation. The elements of a basic xTEDS are shown in Figure 6. The xTEDS are embedded into each device or application that supplies data, provides services, or accepts commands. For the early incarnation of the idea, the xTEDS resident in an appliqué sensor interface module (ASIM) that provides a mechanism to output (and input) data generated (consumed) by the actual hardware component or using a test by-pass mode, output data that is generated by a hardware in the loop (HWIL) simulation.

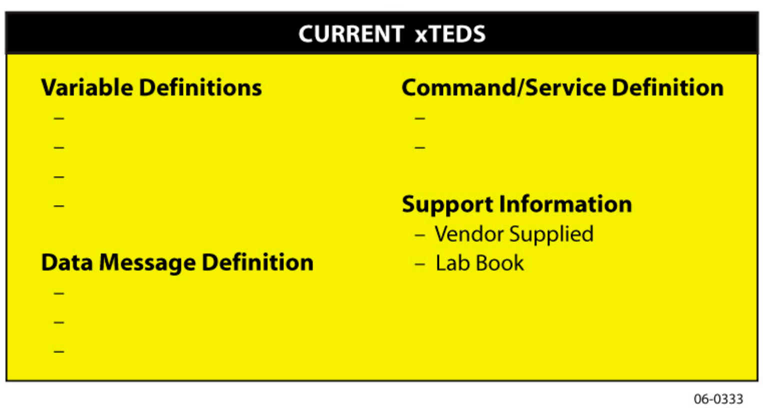

Figure 6: The basic elements of an XML based xTEDS creating a run-time parsable ICD.

IV. The Future of PnP for GN&C

This PnP architecture, described above with respect to implementing a GN&C system, might also include additional extensions that can further increase responsiveness and performance. For example, by extending the test by-pass concept to the operational environment, the device can be operated as a real piece of hardware, as an entirely software simulation, or partitioned with some real and some simulated behavior. These two concepts are illustrated in Figure 7. As stated previously, the attitude knowledge and control of the vehicle is directly related to the quality of the available information. Through on-board simulation and the synthesizing of mixed information, the ultimate quality of the inputs can be optimized. Now the extended xTEDS elements, including simulation and synthesized data, are overlaid on the basic concept and shown in Figure 8.

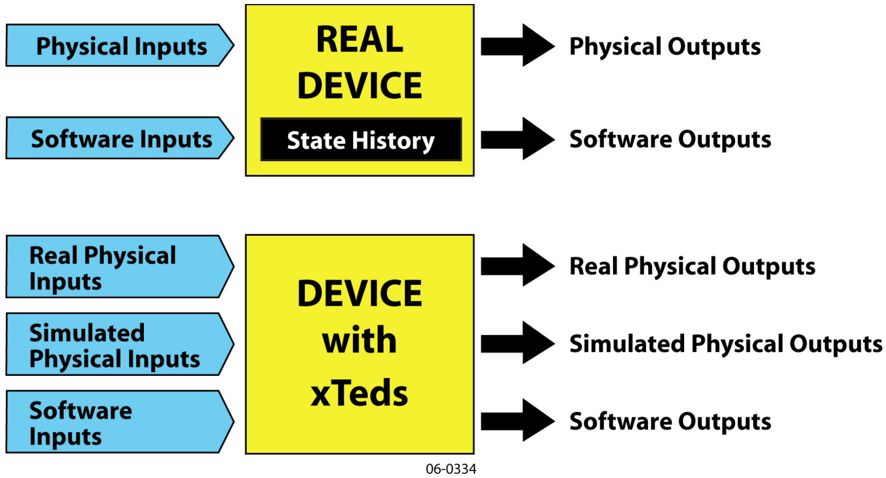

Figure 7: Extending the test by-pass concept to incorporate real device data, simulation data, or synthesized composite data.

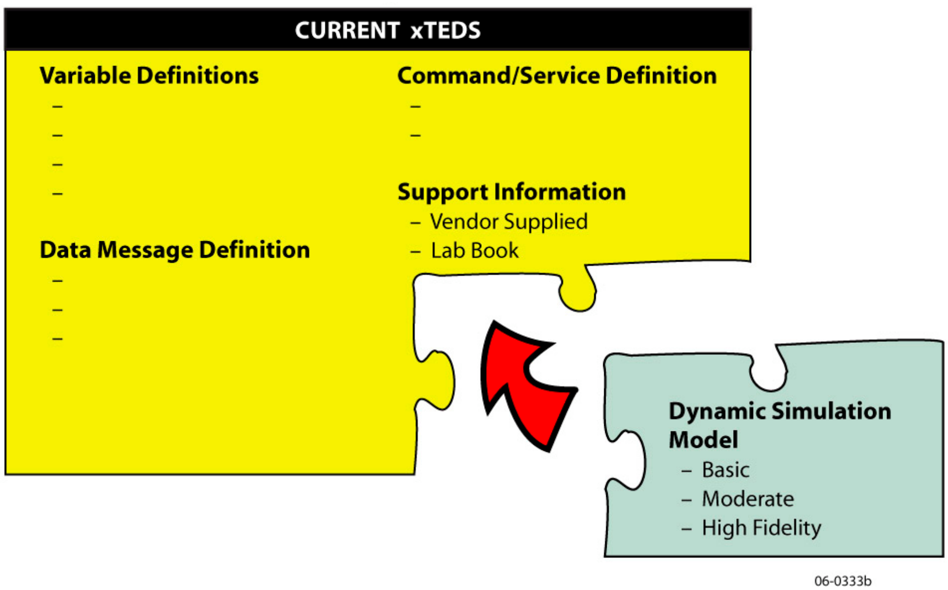

Figure 8: The future xTEDS format created through overlaying the enhanced approach to test by-pass on the basic concept.

Extension of the xTEDS concept to explicitly support the key modeling and simulation activities of spacecraft GN&C development holds the promise of significantly reducing the cost and schedule requirements. Five, often separate, but typically required simulation activities have been identified as shown in Figure 9. Each can have its own fidelity and bandwidth requirements. If, as part of the xTEDS, simulation support was inherently provided, each of the following types of activities could be support transparently, without modifications within the software:

– Transfer functions of various levels of fidelity
– Parameter requirements
– Input data requirements
– Matlab™/Simulink™, STK™ or other platform compatibility

Then, at each stage in the GN&C development, simulations could be auto generated, or at a minimum, considerable time and effort could be saved. This automatic simulation generation capability is already in use for other applications. It will be necessary for spacecraft GN&C in order to achieve advanced PnP capabilities such as adaptive onboard filter generation which typically requires a high-fidelity model of the system dynamics that is updated as the configuration changes.

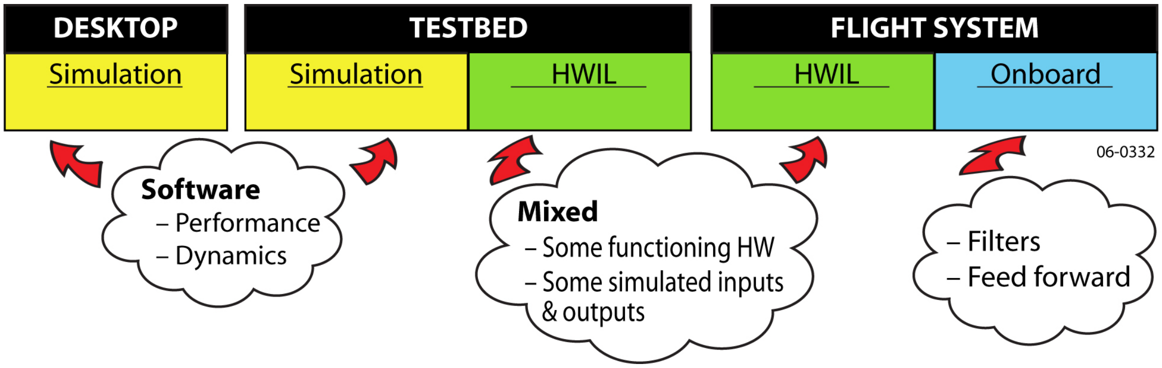

Figure 9: Simulation Activities were supported at all level of development and deployment will lead to faster turn-around and more effective design.

In addition, the transition from "good-enough" to high-performance in the context of low-cost, responsive missions will necessitate auto-generation of high-fidelity simulations and filters/estimators. This is currently much of the work (and cost) associated with typical spacecraft GN&C development. It may, in time, transition the bulk of the development effort for most missions, responsive or not, from getting the simulations right to more value-added design activities. For some missions development and deployment from Matlab™ or other similar environments would become feasible. Additionally, by including simulation as an intrinsic part of the on-board flight software, environmental models can be executed simultaneously with experiencing the effect being modeled, and thus validated and/or modified if necessary.

Figure 10 provides an illustration of how PnP GN&C capabilities could fit into Christensen's disruptive technology framework[2]. Initial forays to demonstrate feasibility would provide relatively low performance while validating the prospect of significant cost reductions. As it becomes accepted and extended, performance and utility ramp-up while development and implementation cost and schedule are driven down significantly for all but the most unique and challenging missions.

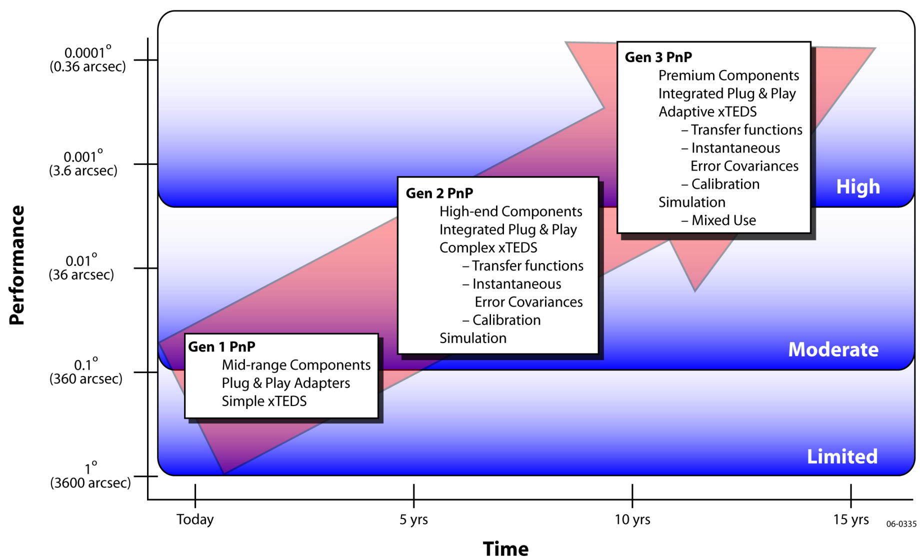

Figure 10: PnP GN&C as a Disruptive Technology will require initial investment but should yield tremendous payoff. [3]

---

[2] From Christensen's book, "The Innovator's Dilemma," extracted from the website: http://web.mit.edu/6.933/www/Fall2000/teradyne/clay.html

[3] Originally presented as part of AAS-06-034Achieving Responsive Space: The Viability of Plug-and-Play in Spacecraft Development

AIAA Infotech@Aerospace 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

V. Demonstrations and Hardware

AFRL has developed a PnP Testbed to demonstrate PnP concepts and validate new hardware and software as it becomes available. Additionally, a spacecraft dedicated to flight demonstration of PnP spacecraft technologies is also under development. Microcosm and HRP Systems are working with AFRL to support much of the associated GN&C development effort, including software and hardware. The Microcosm team is currently developing both a PnP MEMS IMU and a PnP star camera under Phase II SBIR contracts from AFRL.

VI. Conclusion

In summary, as the term operationally responsive space (ORS) continues to be accepted in the spacecraft industry, PnP also becomes a necessary component. Many organizations and disciplines are evaluating what exactly PnP should be for future implementations. This paper has described a vision for spacecraft GN&C PnP that is consistent with the concepts and approach being embraced by AFRL, while extending that approach to a potentially larger paradigm that includes simulation for development and test. The goal of the extension is not just to improve responsiveness with respect to build-time, but also to allow for a mechanism that will enable rapid design and development resulting in a flexible and robust system design transitioning to “better than” vehicles rather than “good enough” vehicles.