iPAS: AES Flight System Technology Maturation for Human Spaceflight

William L. Othon[1]
NASA/Johnson Space Center, Houston, TX, 77058, USA

In order to realize the vision of expanding human presence in space, NASA will develop new technologies that can enable future crewed spacecraft to go far beyond Earth orbit. These technologies must be matured to the point that future project managers can accept the risk of incorporating them safely and effectively within integrated spacecraft systems, to satisfy very challenging mission requirements. The technologies must also be applied and managed within an operational context that includes both on-board crew and mission support on Earth.

The Advanced Exploration Systems (AES) Program is one part of the NASA strategy to identify and develop key capabilities for human spaceflight, and mature them for future use. To support this initiative, the Integrated Power Avionics and Software (iPAS) environment has been developed that allows engineers, crew, and flight operators to mature promising technologies into applicable capabilities, and to assess the value of these capabilities within a space mission context. This paper describes the development of the integration environment to support technology maturation and risk reduction, and offers examples of technology and mission demonstrations executed to date.

Nomenclature

AAE	=	Avionics Architecture for Exploration
AES	=	Advanced Exploration Systems
AR&D	=	Autonomous Rendezvous and Docking
ARM	=	Asteroid Redirect Mission
ARV	=	Asteroid Redirect Vehicle
CAE	=	Common Avionics Enabler
CCSDS	=	Consultative Committee for Space Data Systems
CFS	=	Core Flight Software
DDS	=	Data Distribution Service
DOUG	=	Distributed On-Orbit Ubiquitous Graphics
DRO	=	Distant Retrograde Orbit
DSH	=	Deep Space Habitat
DSN	=	Deep Space Network
DTN	=	Delay Tolerant Network
EDGE	=	Engineering DOUG Graphics Environment
EOL	=	Electro-Optics Lab
DSNet	=	Distributed Simulation Network
ECLSS	=	Environment and Crew Life Support System
F.F	=	Flight Deck of the Future
FSW	=	Flight Software
GN&C	=	Guidance, Navigation, and Control
ICD	=	Interface Control Definition
iPAS	=	Integrated Power, Avionics, and Software
JSC	=	Johnson Space Center
JPL	=	Jet Propulsion Laboratory
KSC	=	Kennedy Space Center
MBE	=	Model Based Engineering

[1] Deputy Chief, GNC Development and Test Branch & iPAS Project Manager

MBSE	=	Model Based System Engineering
MMSEV	=	Multi-Mission Space Exploration Vehicle
MPCV	=	Multi-Purpose Crew Vehicle
NASA	=	National Aeronautics and Space Administration
OTF	=	Operations Technology Lab
PTL	=	Protocol Technology Lab
RDL	=	Rapid Development Lab
REST	=	Representational State Transfer
RF	=	Radio Frequency
STOS	=	Star Tracker Optical Stimulator
SysML	=	Systems Modeling Language

I. Introduction

The NASA Advanced Exploration Systems (AES) projects are tasked with developing key capabilities to be used for future missons. This *capabilities-driven* approach will identify the engineering and operational technologies that must be developed in order to advance human spaceflight beyond Earth orbit. However the AES technologies must be demonstrated *within a mission context* that subjects the proposed flight systems to realistic environmental and operational stimuli. These disparate technology teams often develop products independently, but must demonstrate the ability to integrate together to meet challenging mission objectives.

In order to demonstrate successful technology maturation, system integration, and mission operations, the Integrated Power, Avionics, and Software (iPAS) environment was developed for engineers to: 1) evaluate new technologies for human spaceflight, 2) efficiently integrate and mature technologies into capabilities within a hardware/software/operations environment, and 3) encourage engineers to apply advanced techniques for system design, integration, and test. This environment establishes the infrastructure to incorporate and test technologies as efficiently as possible, through careful design of interfaces within and external to the spacecraft. One goal of iPAS is to *templatize* the test environment functions to the point that technologist can easily and cheaply integrate systems for evaluation.

Along with this technology maturation, engineers and operators must consider new ways of performing system design, development, and integration. The systems required to advance humans beyond Earth orbit will be complex, and will required advanced engineering techniques to make development and integration as efficient and cost effective as possible. The iPAS environment is a proving ground for the development of these new design and integration techniques for vehicle design, integration, and test. By enabling vehicle design and test campaigns on the order of months, people not only apply the techniques in meaningful ways but create timely metrics on which the process can be judged. Tight development cycles allow for identification of processes that work, and refinement of others. iPAS is an environment with sufficient room-to-learn and without heavy project schedule requirements. Early inclusion of operations teams also provides a way to evaluate the “operability” of technologies, and is an important way to establish verification of performance within the community.

In this paper, the features and capabilities of this technology maturation environment will be discussed. Also, the results of successful mission demonstrations with flight-like systems will be presented. These missions include 1) a crewed vehicle inspection of a remote asteroid, 2) the autonomous docking of the Orion capsule with a waypoint station at the Earth/Moon L2 point, and 3) prototype crewed missions in lunar proximity. Engineering technologies tested include processor evaluations, software frameworks and services, deep space command and telemetry infrastructure using Delay Tolerant Network (DTN) standards, wireless sensor technologies, displays and controls, and interfaces with ECLSS systems. For these tests, elements of KSC Launch Control, JPL Deep Space Network, and JSC Mission Operations were included in a real-time integrated distributed test.

II. Technology Maturation on a Budget

New technologies and derived capabilities must be developed to support future space missions far from Earth orbit. But not only can these technologies be expensive to develop, they are also difficult to mature to the point where future project managers will accept the risk of including them. And these technologies must be integrated with other systems and with operations to show maturation towards successful mission execution. NASA must find ways to make this maturation and integration process less costly but still succeed in creating needed capabilities.

Ground testing will never fully satisfy the requirements for technology development, but with flat budgets and access to the space environment relatively rare, these environments can offer an important opportunity to identify and evaluate new ideas. This assessment will help identify the solutions that warrant more time and resources. To improve the fidelity of ground system testing, several key elements were considered.

A. Mission-agnostic Environment

NASA is defining a robust human spacecraft program, and many proposals for future missions are being considered. As these mission are being formulated, engineers and technologists must consider the variety of objectives that NASA will propose for advanced human spaceflight. So technology maturation must be agile enough to accommodate different objectives. While the mission might change, many of the technologies required to support long-duration human spaceflight are similar across the various objectives. iPAS has enabled an agile integration and test environment that can quickly focus on different missions requirements and destinations, and allow *closed-loop* mission demonstration including vehicle hardware and software systems, crew interfaces, and ground operations. The fidelity of the components may vary from simple off-the-shelf elements to “path-to-flight” subsystems, but the integration across the mission context can be demonstrated to show value.

B. Integration Rhythm: Project Formulation, Engineering, & Operations

Ultimately, the process of executing a project and delivering product is the most important aspect of any engineering endeavor. So when developing an environment where technologies can be identified and matured, this project execution and system engineering dimension must also be demonstrated in some tangible way. The requirement to deliver and apply product can help quickly identify issues and discern design errors.

Within the iPAS environment, tests are conducted on roughly six month boundaries. At the beginning of each test campaign, a review is conducted across the disciplines and collaborating projects to determine the test objectives to be achieved and the system components to be integrated. Of course, each AES project will have an independent schedule for product delivery. But through communication across project boundaries, synchronization points are found to support integrated tests.

Specific, tangible products are selected based on inputs from several groups, including Engineering discipline teams, AES projects, Operations teams, and others. Then throughout the delivery and integration phase, issues are addressed and changes made to the product as required. The team learns to deal with problems, to assess them, and to work around them while still maintaining the goal of delivering product at the end of the phase. Through this consistent rhythm of design, integration, and test, the team becomes trained in the processes required to evaluate and deliver capabilities. And the focus remains consistently on implementing and evaluating product using data from test.

Another important aspect of technology maturation is the inclusion of operations considerations early in the design process. Both on-board and on the ground, humans will interact with the vehicle and its subsystems to execute mission objectives. And as the distances from Earth increase, the ways in which crew and ground respond to issues will change. Increased on-board autonomy will be needed when communication with mission control is delayed by many minutes, and issues must be addressed quickly. Ground controllers must learn to account for delay, and use prognostics and other data inference approaches to identify issues as early as possible. The iPAS environment offers an excellent location for this vehicle/crew/ground interaction to be evaluated.

C. Technology Risk Buy-Down

By the time project managers are focused on delivering spacecraft for human exploration, the opportunities to create and integrate new technologies are often lost due to the risk of including them. For technologies to mature to useable capabilities, engineers must look years ahead and predict what will provide the most value for future project manager to succeed. One aspect of the AES Program is to start this identification process, to help the future projects incorporate new technologies.

By supporting early integration and test across disciplines, iPAS can help provide tangible metrics on which decisions can be made. While the specific hardware is often not flight-ready because of the expense, many of the systems are "path to flight" and are in-family with flight components. Subsystem emulators are used when actual systems cannot be co-located, and a distributed network of laboratories has been established to access capabilities wherever they may be. Further, the technologies are demonstrated within a mission-realistic integrated environment, so test results are meaningful from a NASA perspective. These meaningful test metrics can be used to convince managers of value.

The agility of the environment allows components to be interchanged while minimizing the impact to integration within the testbed. So for instance a new type of flight processor can replace existing capabilities, without changing the rest of the integration environment. Thus, consistent metrics can be applied that show the value (or cost) of including the new technology within the system. The environment provides an objective baseline from which judgments can be made.

D. Parallel Development with Ad Hoc Integration

Technologies are often developed independently, within domain-specific laboratories or environments. In the case of certain AES projects, teams are responsible for creating a specific capability or product, such as software, avionics, power, or crew systems. The application of broad system level requirements too early in this development process may hinder the freedom to innovate and explore possibilities. Eventually however, the technology must be integrated with other products and with mission operations elements to demonstrate value. And it is important for the technologists to understand integrated performance. If the effort to integrate is onerous or costly, then engineers will hesitate when considering when to try system integration. While there is clear value with integration, this work cannot impact project milestones. iPAS is designed to mitigate this integration effort. In fact, if done properly, the integration *enhances and improves* project deliveries by demonstrating value.

The iPAS environment is designed to consider *ad hoc* integration, by using clearly defined and open integration standards. Thus projects can consider product development knowing that the effort to bring systems together has been mitigated. This mitigation has several dimensions, including 1) data networks that bring systems together both locally and from distributed locations, 2) interface standard such as those specified by the Consultative Committee for Space Data Systems (CCSDS) and other organizations, 3) a testbed layout that enables access to power, data, and other services throughout the area. The iPAS test environment includes several data networks both within the lab and to remote assets across the country. These networks can allow *distributed, integrated* tests with systems at various locations.

III. An Agile Technology Maturation Environment

At its core, iPAS is a hardware/software/operations integration environment. The environment is designed to enable end-to-end vehicle capabilities demonstration within a mission context. Not all elements of this environment can be flight-ready, since the costs for such a system would be beyond our current budgets. Additionally, the constraints and processes associated with handling true flight equipment could make evaluations difficult and more time consuming. But by judiciously selecting hardware and systems that represent flight components, many capabilities can be evaluated and value can be identified. There are several components of the iPAS environment that enable technology maturation.

A. Vehicle Technologies – The Iron Bird

Within spacecraft development teams, the term *Iron Bird* is understood to be the location where vehicle components are delivered and assembled together to understand integrated performance. Often the iron bird has no outer vehicle shell, and the components are exposed to allow easy access for examination and data extraction. This environment can be invaluable to understand and expose the troublesome integration issues that often are only revealed during system assembly. This environment is also ideal for evaluation of specific types of capabilities, to see the effect on the system performance.

Within iPAS, the Iron Bird represents the closest representation of vehicle systems that budget and test objectives allow. In some cases, elements are path-to-flight. In others, emulators are used to represent the systems of interest. Several vehicle components and systems have been integrated into iPAS, including:

1. Flight software frameworks and applications, on real-time and non-real-time operating systems
2. Redundant flight processors
3. On-board data networks

4. Crew Interfaces, Displays, and Controls
5. Wired and wireless vehicle sensors
6. Power generation and distribution emulators
7. Cold-gas propulsive systems
8. Vehicle communication hardware such as Software Defined Radios and 802.11b Wireless
9. Integration with ECLSS and other domain-specific test labs at remote locations

For crewed exploration of space beyond Earth orbit, the human/system interface is a critical design driver. As the distance from Earth and associated communication delays increase, the crew and vehicle must operate in a more autonomous fashion from ground controllers than has been required historically. The Flight Deck of the Future (F.F) is a development environment managed by Helen Neighbors/JSC and Christie Sauers/JSC within the Avionics Architecture for Exploration (AAE) AES project, whose focus is to identify and develop these crew interface technologies, and demonstrate the integration with other avionics and vehicle systems components. The Iron Bird in iPAS has vehicle network connections to F.F, and can allow for early integration of interfaces and vehicle systems. In this way, new technologies and capabilities can be evaluated with crew in the loop. As will be shown later, telemetry and commands will also be published to ground systems to consider interaction with flight operators as well.

Physical Iron Bird elements are augmented with simulation models of vehicle components as required, to express signals and stimuli that might not be possible otherwise within a ground system. For instance, simulations are used to model range sensors for vehicles many kilometers apart. The sensor model is connected to the vehicle data network, and responds to the avionics and software architecture as if the hardware were present.

B. Testbed Technologies – The Iron Nest

As the Iron Bird represents vehicle systems, the *Iron Nest* is the testbed environment within which the vehicle systems are matured and evaluated. The key requirement for the Nest is to support technology integration as efficiently as possible, but within a mission context that shows value towards achieving NASA goals. The Iron Nest is established on several enabling technologies.

Within iPAS, several test areas have been created to support independent but associated development teams. These testbeds are templatized to provide support and services needed to help with integration and to conduct tests. Power and data network elements are prepositioned throughout the testbed, to make subsystem delivery and integration as easy as possible. Services are provided, so that technology teams do not have to use precious resources on infrastructure.

Test orchestration is the ability to control and manage test execution within one testbed, and across multiple testbeds and laboratories as necessary. For iPAS, a software system called *mREST*[1] is used to provide test coordination. Developed by METECS and based on the Representational State Transfer (REST) architecture, mREST allows the various elements of testbed control to be commanded and monitored from a central location, or even from a web browser on a tablet. The mREST architecture is flexible enough to support integration of different systems and technologies, and provides the means to execute and monitor tests. Because of the data configurability, mREST can be tailored and applied to different projects and teams.

An important element of technology evaluation is the simulation of the operational environment. The fidelity of this simulation is especially important for human spaceflight development, since access to the actual mission environment is rare (if not completely unavailable!). Within iPAS, the Trick Simulation Environment[2] is used to provide this simulation. Created by a team lead by Alex Lin/JSC, Trick has been used at JSC for over 20 years, and has supported Shuttle, Station, and Orion programs. The simulation is capable of modeling the dynamics of spacecraft and celestial bodies in Earth proximity and throughout the Solar System. The simulation also integrates models of vehicle systems not physically present within the testbed, and reports status and accepts commands on the vehicle network. In addition, 3-D graphical images of the mission are provided by the Engineering DOUG Graphics Environment (EDGE), for both engineering analysis as well as modeling of crew interfaces and out-the-window views. The simulation and graphics capabilities are available within each testbed.

Besides providing verified models with a variety of fidelity settings, the simulation also provides an abstracted interface called *I/O channels* that allows the simulation to interface with test orchestration, vehicle networks, and other elements of the testbed. These channels can use a variety of different protocols depending on the nature of the interface itself.

Other basic support features of the Iron Nest include:

1. Data logging and archival support
2. Data networks and integration within the lab as well as with external capabilities, and
3. Source code and data libraries, available through web interfaces

C. Distributed, Integrated Testing – The iPAS Network

The iPAS environment is designed to integrate the products of teams that work independently, in two important ways: 1) through *co-located integrated tests*, and 2) through *distributed, integrated tests.*

Independent, external areas of capabilities development are known as *Federated Labs,* focused on products within a particular domain. These labs are free from constraint of system integration, although often they may include products from other groups for early partial integration. The federated lab can focus on the value being mined within the domain. And these labs may be geographically dispersed, to engage engineers and experts wherever they happen to be. As these distributed teams work independently, they are included in iPAS test coordination efforts; they prepare themselves for integration with broader system elements at the appropriate time. So the requirements for future integration are at least partially considered early.

Some products lend themselves easily for co-located integration. For instance, software can be delivered and loaded on local computers. Many avionics and other systems can be delivered into a lab as well. During the project formulation phase, these products are identified and plans made for physical integration into the lab. And as discussed above, the Iron Bird testbed is designed to assist with the integration process by providing clear interfaces and support services.

Many capabilities and test environments are distributed in various labs and area, and do not lend themselves to colocation. In fact, federated labs often have unique, specialized environments that are vital for testing future vehicle systems. For instance, Environmental Crew and Life Support System (ECLSS) chambers at NASA/Johnson Space Center (JSC) create environments for testing of ECLSS capabilities. For cases where co-location is not feasible, a distributed data network has been developed to allow distributed but integrated testing to be performed. The existing iPAS networks include:

1. Data connectivity across the various testbeds within the iPAS environment
2. Fiber network connections to a large number of specialized test environments across JSC
3. A multi-center data network, called Distributed Simulation Network, or DSNet, which supports data exchange and mission execution involving several NASA centers

Latency must be considered when performing integrated, distributed tests. Of course, vehicle performance requirements that call for microsecond response times cannot be evaluated using a geographically distributed test environment. The test-induced latencies would invalidate any results. There could also be avionics-applied requirements for data command and response time making distributed testing difficult. Control systems often have transport lag requirements, to ensure the time between sensed perturbation to actuation of appropriate control response is optimal.

However there are other types of test where tenths of seconds to even multi-second latency can be tolerated. For instance, data interface testing and command response of loosely coupled on-board systems can be tested. Testing between vehicle and ground systems can also be evaluated. In some cases, latency must actually be added into the test, to simulate telemetry from spacecraft multi-light-seconds away from earth. And there is value to integrate distributed high-fidelity representations of vehicle subsystem components within their native development environments. Subsystem emulators are and should be exchanged between groups, but often cannot reflect the level of fidelity that the native development environments have. So connecting these high-fidelity labs together to do latency-tolerant, integrated tests has value and can be architected.

For any type of distributed test, the test objectives must be clear, and the impacts of any test-system-induced latency must be understood. While vehicle verification will not be performed in a distributed manner, history has shown that important integration work can be done in the presence of latency to identify design issues and data interface problems earlier in the project lifecycle.

D. System Engineering Methodology – Model- and Data-based Engineering

An important aspect of the technology maturation environment involves the identification and promotion of advanced methods for system engineering, integration, and test. Historically, developing or modifying an operational vehicle can be very costly. Even during design, the effort to evaluate value for proposed changes are often outweighed by the inability of projects to be agile and cost-effective about integration. And once the system is performing mission objectives, nominal and off-nominal conditions not considered during design might present themselves, and require a system engineering response.

The iPAS environment is ideal for exposing engineers to modern approaches to system engineering and integration. Because products are integrated across system domains, and integrated tests are conducted every few months, the cycle of integration and test is repeated often. With each cycle, engineers learn about how to specify designs and prepare them for test. Also, with test results created within iPAS, the designs and models are quickly evaluated against real-world results and can be corrected and improved. Thus the model library becomes better quality for the next cycle of development.

The iPAS team is pursing the use of Model Based Engineering and System Engineering approaches (MBE/MBSE). Domain engineers such as avionics or power teams are learning to express designs using System Modeling Language[3] (SysML) formalisms. By describing designs in a machine-parsable format using SysML, the designs can be mined and evaluated. And the closer these models are to the actual designs of the engineers, the more valuable they become for vehicle analysis. Useful products can be created or extracted directly out of the models as well. These products can include mass tables, power requirements tables, and even software interface control definitions (ICDs). The MBSE approach has been pioneered at JSC by Lui Wang/ER and a team of engineers from Tietronix Software Inc.

IV. AES Technology Demonstrations

As has been discussed, NASA has established a set of projects whose purpose is to develop technologies to enable future human spaceflight beyond Earth orbit. These AES projects develop independent milestones and products, but also plan for integration sync points that demonstrate how capabilities work together to produce a functioning system.

The goal for each iPAS test is to develop a *mission context* to help with evaluation of integrated system capabilities, one that is aligned with current strategies for human exploration. The agility of the environment is demonstrated with each test. As the missions change from year to year, both the AES technologies applied (the Iron Bird components) and the test environment (the Iron Nest elements) are reconfigured and applied to the new mission thus demonstrating agility.

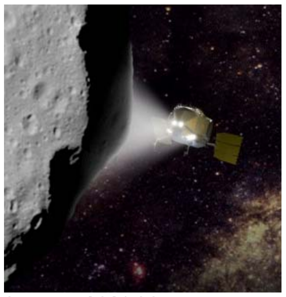
A. Creating the Coalition For Technology Advancement

People are the most important aspect of technology maturation and spacecraft development. Assembling the team of engineers and operators is the first critical component to the success of any endeavor. Often this team is assembled naturally during project formulation, to create one particular product such as a spacecraft or a science payload. However in our current environment of technology development, the various teams that build capabilities are often separated into individual projects with specific but independent products. The key to advancing technology affordably is to align these projects in such a way that integrated demonstrations and tests are performed, while directly supporting the achievement of milestones within each of the projects themselves. In that way, the projects show value for their stakeholders, and the capabilities are demonstrated to work together to fulfill NASA goals.

In early 2011, as the concepts for the iPAS environment were being developed, a team was formed that included representatives from several AES projects as well as Engineering and Operations organizations. The team is called iPAS Pathfinder, and was intentionally formed across different technical communities and NASA centers to ensure broad application of technologies and operational perspectives. This team is involved in definition of test objectives, and also in the selection of the specific products to be demonstrated during a test. They also support the selection of the mission context, to maximize the value of the test.

This coalition is enabled in part by the iPAS integration environment discussed earlier. Teammates leverage off the data networks within JSC and across NASA centers, and are directly involved in tests. During tests, video conferencing technology is used to allow direct interaction with teammates wherever they are. This face-to-face interaction enhances test operations and more completely involves remote participants.

B. Asteroid Visitation


Source: NASA/JSC

In 2011, one mission to expand human presence in space proposed a crewed vehicle visiting an asteroid between Earth and Mars. This mission concept was considered to be a stepping stone towards an eventual mission to Mars.

One of the AES projects, lead by Mike Gernhardt/JSC, envisioned an excursion vehicle called the Multi-Mission Space Exploration Vehicle (MMSEV) that would allow humans to encounter and explore the asteroid. The MMSEV was not the transit vehicle that would enable travel to the asteroid, but the crewed craft that would undock and explore the asteroid on arrival.

Several other AES projects are developing products critical to the advancement of human spaceflight. The Avionics Architecture for Exploration (AAE) project, lead by Jim Ratliff/JSC, is identifying and developing key products in the areas of flight processors, vehicle networks, displays and controls, and communication architectures. The Core Flight Software (CFS) project, lead by Lore Prokop/JSC, is maturing an existing software framework for possible use in future human spaceflight. The products from these AES projects were instrumental for developing an integrated test within iPAS.

To start the planning phase of testing, the iPAS Pathfinder team conducted a workshop to help provide focus on tangible products. The Asteroid Visitation mission was selected for the first iPAS integrated test, and the team executed a Project Formulation phase to specify exactly what components and products would be integrated to conduct the mission. Representatives from the various AES projects as well as Engineering and Operations technical domains across NASA met, and elements from each discipline were identified by the team. This formulation phase was important to ensure there was agreement within the team about what could be achieved in the short timeframe that was available, in this case about four months from concept to test execution. Defining roles and responsibilities and then getting agreement on a common vision are two critical elements to successfully conducting these integrated tests. Several groups and capabilities were integrated, as described in Table 1.

A key product that is often ignored during engineering development is the training involved to *learn how to produce and integrate products*. During the summer of 2011, the team met weekly to discuss product development, to plan for incremental integration, and to deal with integration issues. Small integrated tests were planned both within Federated Labs where technologies were being developed, and also in iPAS where integration across systems was performed. The process of communicating issues and driving towards a product as a group can be challenging. The iPAS environment provided a unique environment to teach the team how to succeed.

On 29 September 2011, the first integrated test was formally conducted in iPAS. During this test, the MMSEV vehicle undocked from a representative transit vehicle, then autonomously moved among a series of waypoints identified around the asteroid. During the mission, the MMSEV would advance and stop at each waypoint. Using crew displays within F.F, the MMSEV could be commanded to proceed to the next waypoint. In this way, the vehicle could circumnavigate the asteroid and conduct an inspection.

Table 1
Subsystem Components Integrated for Asteroid Visitation Demonstration

Subsystem Name	Affiliation	Subsystem Lead	Description
System Engineering and Integration	iPAS Team	Bill Othon/JSC	iPAS Development and Integration Lead
iPAS Test Operations	iPAS Team	David Fletcher/JSC	Test Operations Lead
Test Automation	iPAS Team	Chatwin Lansdowne/JSC and Pat McCartney/METECS	Test Automation Scripts development
Flight Processor & Data Networks	Avionics Architecture for Exploration (AAE) AES	Greg Hall/JSC	Selection of the flight processor for the vehicle
Core Flight Software (CFS)	CFS AES	Lore Prokop/JSC	Software framework for vehicle FSW
Guidance Navigation and Control Application	GNC Development and Test Branch/EG2	Jen Madsen/JSC	GNC algorithms, leveraged from work by Paul Gesting/Odyssey
Simulation Environment and 3D Graphics	Simulation and Graphics Branch/ER7 and iPAS team	Toby Martin/JSC, Frank Graffinino/Metecs	Simulation Integration lead, leveraging work by Zack Crues/JSC, et. al.
Flight Software Integration and Real-Time Systems	GNC Development and Test Branch/EG2	Teming Tse/JSC and Tim Runkle/JSC	Production of Flight Executable
Power System	Power Branch/EP5	Mike Salinas/JSC and Cindy Situ/JSC	Emulators of Power Generation and Distribution Systems
Propulsion Systems	Propulsion Systems/EP4	Joe Durning/JSC	Cold-gas jet system (4 active jets)
Displays and Controls	AAE AES	Mary McCabe/JSC and Christie Sauers/JSC	Crew Displays and Control interfaces
Command and Telemetry	AAE AES	Laura Hood/JSC	On-board Telemetry and Command architecture
Deep Space Network	JPL Protocol Technology Lab (PTL)	Leigh Torgerson/JPL	Modeling of the Deep Space Network and support for Delay Tolerant Network
Mission Operations	Operations Technology Facility (OTF)	Tom Rich/JSC	Development of Standards for future mission ops
Launch Control	Avionics and Software Group	Mike Peacock/KSC	Interfaces with Launch Control systems
System Administration	iPAS Team	Gigi Mathew/Jacobs Engineering	iPAS System Admin
Network Design	iPAS Team	Omar Baltaji/Jacobs	Testbed networks

Vehicle elements had been integrated together:

- The selected flight processor was loaded with Core Flight Software and the GN&C software algorithms used to fly the spacecraft autonomous around the asteroid.
- A vehicle Ethernet network was created, that connected the flight computer with the power system, displays and controls, and other vehicle subsystems.
- A 4-jet cold gas system was constructed, that responded to control system commands sent from the flight computer to the power system, and pressure transducer data was then sent back information to the flight computer.
- Displays were developed to provide situational awareness for crew during the mission and supported commands to the flight computer for mission moding.
- On-board telemetry and command was enabled, through services provided by CFS. CCSDS Space Packets were telemetered from the vehicle through the testbed interfaces to external users.

Operations elements were integrated:

- The data interface to the Launch Control Center emulator at KSC was established. During the test, engineers at KSC sent commands to turn on vehicle power, and then next turn on the flight computer. These commands emulated the interaction between the LCC and a vehicle on the launch pad.

- The command and telemetry interface to the JPL Protocol Technology Laboratory (PTL) was established. Using a data publishing capability delivered by Tom Rich/OTF, CCSDS space packets were forwarded from the vehicle network through the DSNet system to JPL, where the Deep Space Network simulation was used to model delay and data drop-out.
- The command and telemetry interface to the OTF was established. Vehicle telemetry processed by the JPL DSN simulation was sent to the Operations facility, where simple emulations of flight operator displays depicted the data. In addition, commands from the ground through DSN and back to the vehicle were also enabled. During the test, operators in the OTF sent a command for a fan to be activated on the vehicle. This command was routed through the JPL facility and forwarded with the appropriate latency onto the vehicle through the flight computer. On receipt of the valid command, the flight computer commanded power to the fan through the on-board power system.

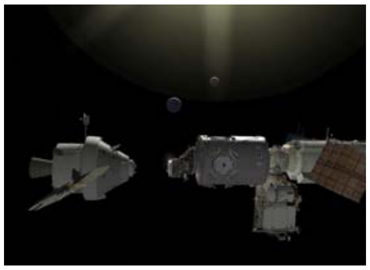
Testbed capabilities were applied:

- The mREST test orchestration scripts were encoded to support test execution from a web browser.
- The Trick environment simulation of the Heliocentric environment was developed, and integrated with the flight vehicle network to support closed-loop vehicle control.
- The EDGE 3D graphics tool was used to depict both out-the-window views and crew monitor emulation, as well as external eye-points used for mission evaluation and situational awareness.

The mission demonstration had successfully shown that the team could quickly assemble products from various projects and integrate them together to achieve value for NASA. The team was awarded a NASA Group Achievement Award to recognize the important work done, not only to produce technical value but also to show that a broad team could be quickly assembled to achieve capabilities maturation.

C. Orion and Waypoint Vehicle

In the Earth/Moon system, points exist that have special gravitational properties. These *Lagrange Points* offer marginally stable locations that minimize (but do not eliminate) the energy required to dwell there. One mission concept proposed to locate a vehicle at the Earth/Moon L2 Lagrange Point, to provide an environment for learning how to live outside the protection of Low Earth Orbit and to support a staging area for the exploration of space. The Orion Multi-purpose Crew Vehicle (MPCV) would perform autonomous rendezvous and dock (AR&D) with this remote station, to shuttle crew and equipment to the Waypoint Vehicle.


Source: iPAS/JSC

This new mission context provided the opportunity to demonstrate the agility of the team as well as the flexibility of the iPAS environment to apply capabilities to a new mission objective. Again the iPAS Pathfinder team conducted an organizing workshop, except this time there were many products that could be leveraged from the previous year. Many vehicle systems had been put in place, software system and repositories were available, and interfaces to local and remote laboratories were already established. However new systems were also created, and are summarized in Table 2.

An important advance for this phase of iPAS development was the creation of an Orion emulator within iPAS. In order to conduct Orion AR&D with Waypoint, new CFS applications were developed. These applications were created using a process of autocoding from the Mathworks Simulink tool, in a way very similar to how the actual Orion FSW is being developed. The autocoded Simulink algorithms were encapsulated into a CFS application, then integrated into the CFS framework on a flight computer.

Physical (non-flight) engineering representations of Orion systems were also added. The cold gas jet system was expanded to include 16 jets, to represent the number available on the Orion Service Module. Crew interfaces were updated to include prototypes of proposed Orion displays, delivered by the Rapid Prototyping Lab team lead by Lee Morin/JSC. Translational hand controllers were added, to allow a pilot to take control from the automated system.

Table 2
Additional Components Integrated for Orion/Waypoint AR&D Test

Subsystem Name	Affiliation	Subsystem Lead	Description
Orion GN&C Applications	GNC Development and Test Branch/EG2	Matt Fritz/Draper Labs	Integration of Orion AR&D Autocode into CFS
CFS Product Line	AAE AES	Steve Duran/JSC	Delivery of the official CFS Product Line
Waypoint Vehicle Interface	Deep Space Habitat (DSH) AES	Danny Carrejo/JSC, Lui Wang/JSC, Dennis Lawler/JSC	Interface between Orion network and Waypoint
Star Tracker	GNC Development and Test Branch/EG2	Steve Lockhart/JSC	Integration of signals from Star Tracker into CFS
Software Defined Radio	AAE AES	Adam Schlesinger/JSC	Modeling Vehicle to Vehicle communication
Wireless Sensors	AAE AES	Rick Barton/JSC and Ray Wagner/Jacobs	Wireless sensors feedback to Flight Computer
Propulsion Systems	Propulsion Systems/EP4	Joe Durning/JSC	Emulation of Orion SM Jets
Orion Displays and Controls	AAE AES	Helen Neighbors/JSC, Max Paddock/JSC, Christie Sauers/JSC	Integration of Orion displays, delivered from the RDL lead by Lee Morin/JSC
Environmental Crew Life Support System (ECLSS)	Crew and Thermal Systems/EC	Andy Dawson/Jacobs	Interface vehicle with ECLSS chamber and ECLSS sim
Processor Redundancy	AAE AES	Greg Hall/JSC	Demonstration of Hot Backup capability
Simulation Environment and 3D Graphics	Simulation and Graphics Branch/ER7 and iPAS Team	Mike Gaboury/Odyssey, Andrew Spenser/Odyssey, Frank Graffinino/Metecs	Simulation Integration lead, leveraging work by Zack Crues/JSC, et. al.
System Administration	iPAS Team	Walt/Wilson/Jacobs	iPAS System Admin

To demonstrate remote distributed testing, a vehicle data connection was extended from iPAS to another building at JSC that housed the Electro-Optics Lab (EOL), managed by Steve Lockhart/JSC. Within EOL, a simple (non-Orion) star tracker was mounted in front of a Star Tracker Optical Simulator (STOS). The star field modeled by the STOS was oriented by the Trick simulation executing in iPAS hundreds of meters away. The field was positioned to display what the star tracker on Orion would observe, based on the simulation dynamics. The physical star tracker would be stimulated by the STOS and would identify the vehicle attitude. A device called a Common Avionics Enabler (CAE), developed by Greg Hall/JSC and Ayman Quddumi/Jacobs, transformed the output of the star tracker from serial to Ethernet output, to allow the device to be connected to the vehicle network. From the CAE, the data was sent to the CFS FSW running in iPAS. This configuration demonstrated the ability to integrate components at remote labs within the iPAS test environment, at the cost of some milliseconds of data latency. For this test, the star tracker data was not integrated closed loop with the Navigation software, but that enhancement is currently under way.

The testbeds were architected to improve integration testing. One example was used for intra-vehicle communication. During the rendezvous portion of the test, the Orion and Waypoint spacecraft would communicate using the Software Defined Radios physically present in the lab; the Radio Frequency (RF) communication was emulated between the radios using a co-axial cable. However when the simulation identified the range below a certain value (simulating a dock), the testbed system would activate a local network switch, and the two vehicles could communicate over a physical Ethernet network emulating the result of a successful docking. In this way, the dynamic nature of data discovery between two independent but docked systems could be evaluated.

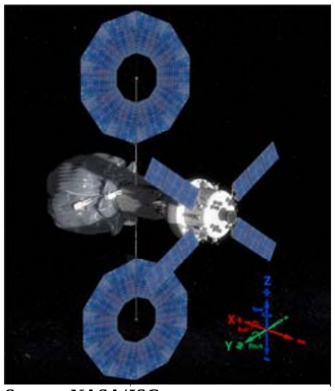
Another advance for integrated testing was the integration with an existing, separate testbed. The Deep Space Habitat (DSH) AES team already had an fully-functional avionics testbed environment, but they used a different protocol for vehicle data exchange called the Data Distribution Service[4] (DDS). For this test, elements of the DSH testbed would represent the Waypoint vehicle. So information had to be passed from the DSH testbed to the new iPAS Waypoint testbed. Danny Carrejo/JSC developed a DDS to CFS bridge, that established this data interface and allowed information to flow between the two distinct groups. This was the first time that independently developed avionics systems had been interfaced within iPAS, and laid the foundation for future collaboration.

On 28 September 2012, the demonstration of the Waypoint AR&D mission was successfully performed within iPAS. For this test, the Orion vehicle was positioned 2 kilometers away from the Waypoint Vehicle. Following an automated timeline, Orion performed a rendezvous burn, targeted for 300 meters away from the Waypoint. After completing this first approach phase, Orion maintained station-keeping at this 300 meter location. After 10 minutes, the Orion continued proximity operations by closing to 10 meters and stopping for another 10 minutes. Finally, Orion closed the final 10 meters until range was identified below 1 meter, at which point the test was concluded.

Within the span of five months (from May to September), the iPAS environment was reconfigured from one mission (MMSEV) to another (Waypoint AR&D). New technologies had been integrated, new team member incorporated, and the mission successfully demonstrated. The team received a well-deserved JSC Group Achievement Award.

D. Asteroid Redirect Mission

A recent NASA proposal envisions the retrieval of portions of an asteroid, or the redirection of a small asteroid, into a stable orbit around the moon. Called the Asteroid Redirect Mission (ARM), the asteroid or fragment would be captured in deep space by a robotic vehicle, the Asteroid Redirect Vehicle (ARV), returned and positioned in a Distant Retrograde Orbit (DRO) around the moon, and then would be visited later by crews aboard the Orion spacecraft. Several important technologies including propulsion and life support systems could be demonstrated in this near-Earth "proving ground", and then applied later for future missions to Mars.



Source: NASA/JSC

This new mission offered another opportunity to demonstrate the flexibility of the iPAS system. Again, elements within iPAS were leveraged, to allow the group to execute a mission demonstration much faster than otherwise possible. To support early mission concept evaluations, a Trick ARM simulation was developed by a team lead by Zack Crues/JSC and delivered to iPAS. This simulation was again modified to interface with CFS, but these interfaces had already been developed so less time was needed for development. New systems that were added are described in Table 3.

Table 3
Additional Subsystem Components Integrated for Asteroid Redirect Test

Subsystem Name	AES Project	Subsystem Lead	Description
Simulation Environment and 3D Graphics	Simulation and Graphics Branch-ER7/JSC and iPAS Team	Andrew Spenser/Odyssey, Frank Graffinino/Metecs	Simulation Integration lead, leveraging work by Zack Crues/JSC, et. al.
GN&C Development	GNC Development and Test Branch-EG2/JSC	Mike Butticoli/JSC, Matt Fritz/Draper	Integration of Orion EM1 FSW autocode into CFS

An important new element was the incorporation of the Orion Absolute Navigation (AbsNav) autocode for EM1, performed by Mike Butticoli/JSC and Matt Fritz/Draper. The Orion AbsNav design was specified in the Mathworks Simulink tool, and then autocoded to C++. The C++ FSW product represents the same software that will be used by Orion, but for this test was integrated into CFS. This ability to integrate the actual Orion FSW allows teams to do early testing and demonstration in a real-time, closed-loop environment. For this test, the Navigation was not integrated closed-loop with the Guidance and Control, but only performed an open-loop estimate of absolute inertial position. Closed-loop integration is planned for subsequent iPAS tests.

In addition, software models of Orion subcomponents were integrated into the Trick simulation by Andrew Spencer/Odyssey. These models improved the fidelity of the Orion emulator in iPAS, by using the same models applied to Orion GNC performance evaluations.

Because of the similarities to the Waypoint mission, few changes were required to the Orion AR&D FSW. The Orion was again placed 2 km from the target vehicle, which in this case was the ARV and Asteroid stack in a Distant Retrograde Orbit (DRO) in lunar proximity. A representative vehicle attitude control system was delivered with the ARM Trick simulation. The Orion successfully executed the rendezvous and dock mission with the ARV/Asteroid stack.

V. Conclusion

NASA has been at the forefront of technology maturation since its inception. Engineers have imagined, designed, and developed amazing capabilities to support scientific, robotic, and crewed exploration of the cosmos. Today with constrained budgets and increasingly complex mission objectives, engineers and operators are forced to consider new and innovative approaches to solving these difficult requirements. The technical areas being evaluated by AES projects are one important component of this strategy. AES teams are identifying and developing key capabilities required to advance human spaceflight beyond Earth orbit.

iPAS supports this technology maturation and capabilities demonstration activity, by creating an environment that enables technology integration and metrics collection within a mission context that highlights value to the agency and beyond. The environment has been architected to encourage innovation and facilitate integration across a wide range of technical and operations disciplines. Local and distributed network capabilities ensure that inputs from engineers can be accepted and included, wherever they are. And the environment supports new way of doing business, in an environment that allows people to learn and to improve as they develop. iPAS is itself an enabling technology, ready to help NASA meet the challenges of the coming decades.

Acknowledgments

This paper summarizes the incredible achievements of members of several AES projects and the iPAS team. The author wishes to acknowledge the following for their contributions: Bill Arceneaux, Omar Baltaji, Lynne Barends, Rick Barton, Dan Benbenek, Matt Benson, Mike Butticoli, Allen Brewer, Danny Carrejo, Edwin. Z. Crues, Lydia Davis, Dan Dexter, Steve Duran, Joe Durning, David Fletcher, Matt Fritz, Mike Gaboury, Christina Gallegos, Jackie Gillman, Monty Goforth, Pete Gonzalez, Frank Graffagnino, Mike Grygier, David Guibeau, Max Haddock, Greg Hall, Kevin Hames, Laura Hood, Michel Izygon, Mark Jackson, David Jih, Scott Kauffman, Chatwin Lansdowne, Steve Lockhart, Jennifer Madsen, Mike Marston, Toby Martin, Lindolfo Martinez, Gigi Mathew, Mary McCabe, Pat McCartney, Ian Mitchell, Lee Morin, Tam Ngo, Sam Nguyen, Hung T. Nguyen, Shira Okon, Mike Peacock, Jason Plumber, Lore Prokop, Ayman Qaddumi, Jim Ratliff, Adam Rawlin, Sue Reno, Tom Rich, Rod Robinson, Tim Runkle, Mike Salinas, Christie Sauers, Adam Schlesinger, Josh Schoolcraft, Brian Schwing, Cindy Situ, Andrew Spensor, Greg Stiggins, Paul Sugden, Kevin Tones, Leigh Torgerson, Victor Tran, Teming Tse, Tim Verborgh, Sharada Vitalpur, Ray Wagner, Lui Wang, Mihriban Whitmore, Jonathon Wilmot, Walt Wilson, Chris Winton, Eric Wolfer, Doug Wong.

References

[1]Lansdowne, C.A., Maclean, J.R., Graffagnino, F.J., and McCartney, P.A., "Automation Hooks architecture trade study for flexible test orchestration," AUTOTESTCON, 2010 IEEE , vol., no., pp.1,3, 13-16 Sept. 2010

[2]Paddock, E., Crues, E., Lin, A., and Vetter, K, “Trick: A Simulation Development Toolkit,” AIAA-2003-5809 AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas, Aug. 11-14, 2003

[3]Friedenthal, Sanford, Moore, Alan, and Stiener, Rick, *A Practical Guide to SysML: The System Modeling Language*, Morgan Kaufmann OMG Press, Burlington, MA, 2009

[4]Van Bruaene, Jan, “Standards-Based Plug-and-Play Data Distribution”, AIAA-2007-2908 AIAA InfoTech Conference and Exhibit, 7-10 May 2007