

# Aircraft IMA Integration Bench

Managing the Challenges of Integrated Modular Avionics

## Introduction

Integrated Modular Avionics (IMA) represents a marked shift in commercial and military aircraft technology. The Airbus A380 and the Boeing 787 are two prominent, advanced aircraft programs that led the commercial push to embrace IMA architecture. The industry's two fiercely competing leaders' embrace of IMA seems to indicate that IMA is here to stay. The Airbus A350XWB as well as a collection of yet-to-be-announced new aircraft programs are moving to IMA, which contributes significant cost-savings to aircraft operators. The commercial and military aircraft industry is not usually characterized by sweeping, wide-reaching technology change but rather by small incremental change. So, the many aspects of new technology insertion associated with IMA represent something significant.

Key elements of IMA include:

1. A distributed architecture where avionics functions are divided into: centrally computed software application(s) and the remotely located End Systems, connected by a high-bandwidth network backbone
2. An "IMA Platform" providing shared computational resource to execute avionics application software
3. A shared, dual-redundant Ethernet network (ARINC-664/AFDX) featuring multi-cast messaging, intelligent switches, and a safety-critical/time-critical network protocol

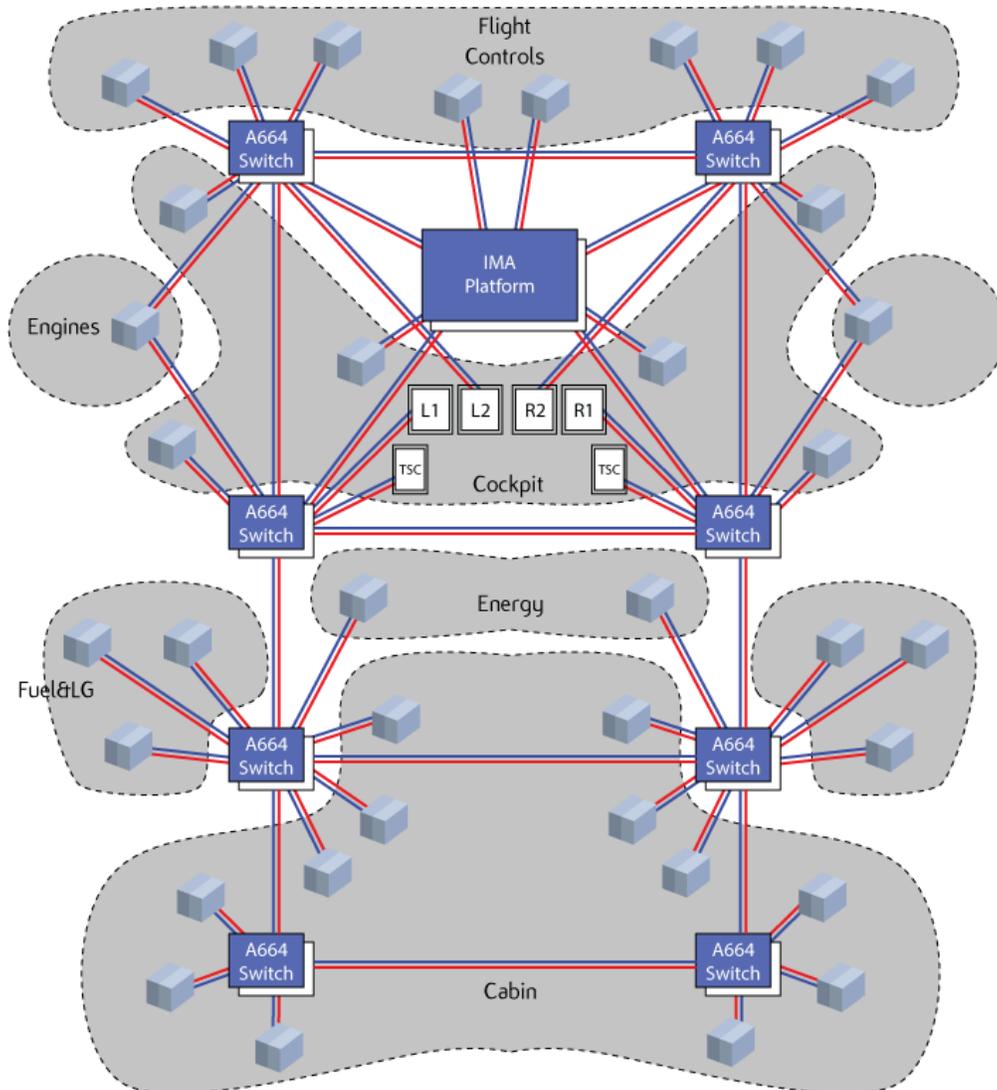
## IMA Architecture

IMA is a departure from the 'federated avionics' architecture. In a federated avionics architecture, each aircraft system is physically firewalled from one another with dedicated computational resources, dedicated cabling, and limited commonality from system to system. Although aircraft systems within a federated avionics architecture share information, sharing is typically a low-bandwidth affair, using legacy ARINC-429 databuses, resulting in many miles of wiring.

The IMA architecture is distributed, where a given avionics function (ex: landing gear extension and retraction) is essentially split into two parts:

1. The software application – the control algorithm, health diagnosis, failure mode actions, etc.
2. The End System – the sensors, actuators, mechanical, electrical, hydraulic components, etc.

These two parts of a single function are separated by distance and connected across a shared high-bandwidth (within the frame of reference of safety-critical aircraft networks) network. By connecting these two parts with a shared network, the amount and weight of aircraft cabling is significantly reduced. The trend in aircraft systems is increasing numbers of sensors and electrical loads. Reduction of cabling between these subsystems in modern, large commercial aircraft, which traditionally have hundreds of miles of copper wire, translates directly into decreased aircraft weight and fuel consumption. Therefore, the IMA architecture offers a path to reduced operating costs of a given aircraft and a more compelling product.

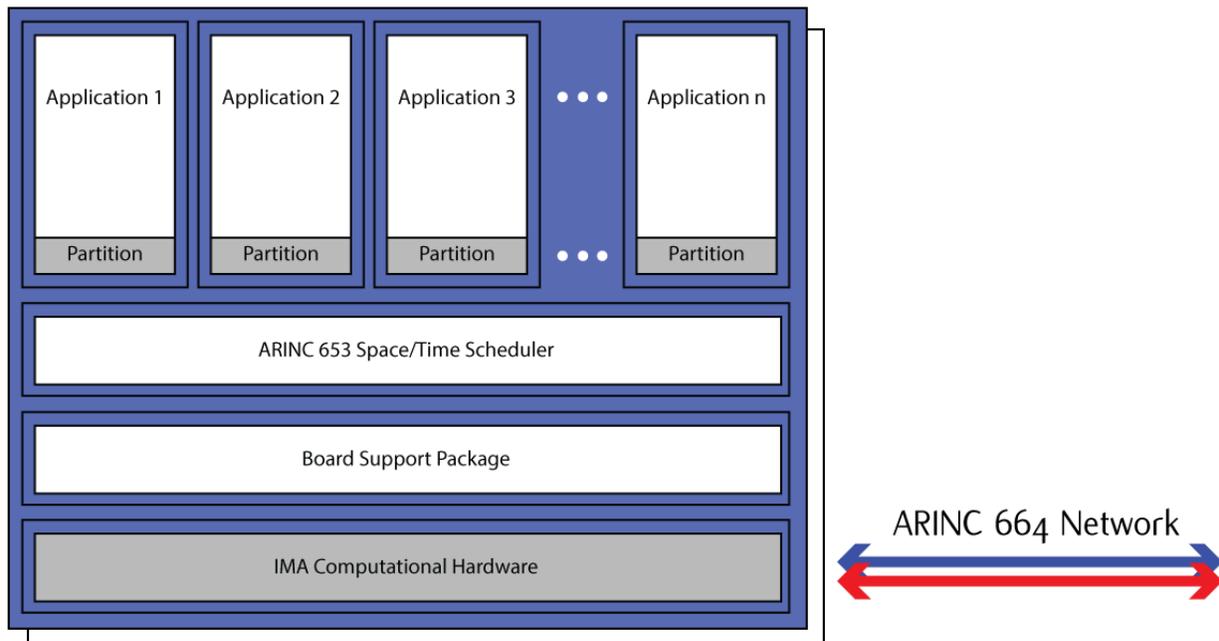


The figure above illustrates an IMA architecture including the centralized IMA platform providing shared computational resources used to execute software applications, a shared dual-redundant Ethernet network (ARINC-664/AFDX) with intelligent network switches, and remotely located End Systems.

### IMA Platform

The IMA Platform provides a shared, system-agnostic computing platform - an impressive feat of safety-critical systems engineering. Robust Operating System (OS) partitioning and application scheduling are key components of the IMA platform. A time/space scheduler ensures deterministic execution of safety-critical applications by providing strict allocation of computing resource execution time slices. IMA uses strict memory partitioning to allocate memory to each application and a Memory Management Unit (MMU) detects and prevents violations. Leading safety-critical RTOS suppliers and notable commercial aircraft programs have embraced ARINC-653 as the leading industry standard for IMA Real Time Operating Systems (RTOS).

## IMA Platform



In contrast to a federated avionics architecture where each aircraft system supplier delivers a dedicated avionics computer to execute the system's software application, IMA-platform system suppliers deliver software applications that will execute on the IMA platform shared computing resource.

As an aside, IMA-architecture aircraft to date have moved some of the software applications to the IMA Platform but not all. For example, the Flight Control Computers and the FADEC (Electronic Engine Controller, Fuel Metering Unit, etc.) applications are typically hosted on dedicated avionics computers as in the federated avionics architecture. However, the aim is to move more and more software applications away from system-specific computing resources to the IMA Platform. In addition, the remotely located End Systems connected to the ARINC-664/AFDX network are intelligent electronic systems that include some amount of software functionality such as local closed-loop control and safety functionality.

Shared IMA platform computational resources offer the added benefit of lower maintenance cost associated with maintenance spares. By reducing the number of system-specific avionics computers and making greater use of common IMA Platform computational hardware, the number of spares held to support a fleet may be reduced. The IMA Platform approach also improves flexibility within the aircraft development program. If, over the course of the program, the computational load of a given application function exceeds initial estimates, the IMA Platform provides the flexibility to reallocate the application to a suitable IMA Platform processor module without need to redesign the aircraft Electronic Equipment Rack (EER) layout, cooling, etc.

### Dual-Redundant Ethernet

The third key element of an IMA-based aircraft is a dual-redundant ARINC-664/AFDX network. ARINC-664 combines the standards-based, low-cost Ethernet technology that evolved in the IT industry with a

deterministic, safety-critical hardware implementation and protocol. ARINC-664 uses the concept of a Virtual Link (VL) whereby messages are transmitted by a single End System and received by one or more End Systems. Network traffic is grouped into VLs and bandwidth is dedicated to each VL using the concept of a Bandwidth Allocation Gap (BAG). The IMA ARINC-664 network implements a cascading star topology with multi-cast messaging by using VL-aware, intelligent network switches. In addition to providing deterministic multi-cast data transmission, the ARINC-664 switch also includes error detection and reporting.

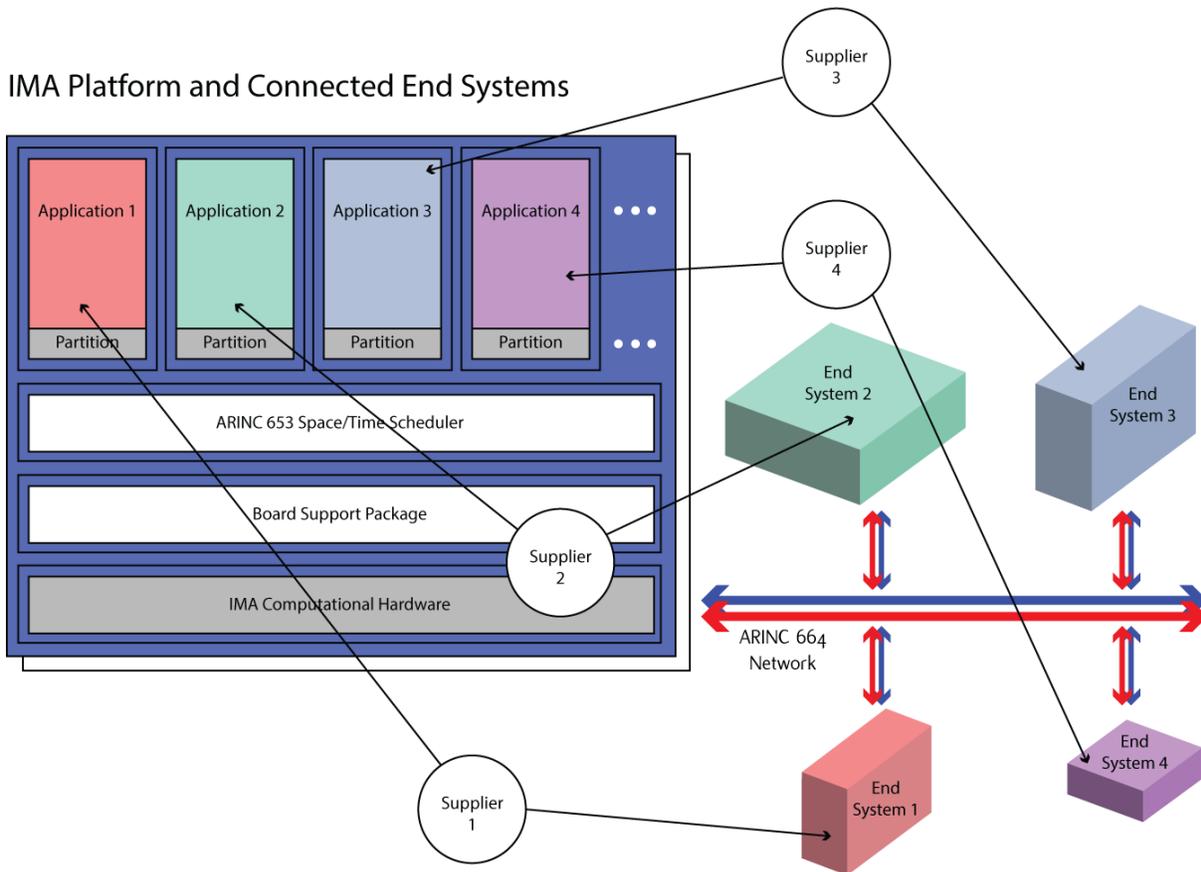
The ARINC-664 network offers one thousand times the bandwidth of the ARINC-429 databus it replaces. This translates directly into reduced cabling, reduced aircraft weight, and lower aircraft operating costs. Because ARINC-664 is based on standard Ethernet silicon and software, it adds tremendous long-term cost savings versus once-competing, specialized technology such as ARINC-629.

## Integrating and Verifying IMA Aircraft Systems

### End Systems and their Avionics Application

There are a few exceptions, e.g. engine manufacturers developing engine control laws hosted on Electronic Engine Controllers and airframers developing flight control laws hosted on the Flight Control Computer, but for the most part, the idea of aircraft system suppliers developing software applications hosted on a third-party airborne computer system represents a revolutionary.

IMA Platform and Connected End Systems



The figure above shows the IMA Platform with four applications, each supplied by a different aircraft system supplier, each executed on shared computational resources, and each communicating across a shared network resource to their respective End System.

With aircraft systems' increasing dependence on software and electronics, and on data sharing, the challenge of integrating these systems and verifying system interoperability has increased. In an IMA-based aircraft this task is suddenly far more challenging. Today's aircraft development program involves aggressive development schedules. To support these short schedules, aircraft system integration and verification depends on using Model Based Systems Engineering (MBSE) techniques to support an evolutionary, or agile development approach using simulation to integrate and verify systems well ahead of access to representative airborne equipment and flightworthy software.

## MBSE Techniques for IMA Integration

Using real-time, hardware and pilot-in-the-loop integration and verification facilities is standard practice for major aircraft development programs. Avionics Integration Facilities, Iron Bird testing facilities, and engineering cockpit simulation facilities are common MBSE techniques and verification methods used by most airframers.

IMA-based aircraft development introduces a new role: The role of the IMA Manager. The IMA Manager, or management team takes responsibility for the allocation of the IMA's shared resources. The IMA Platform (ARINC-653 in particular) and the ARINC-664 network make heavy use of XML tables to specify the configuration of these shared resources. XML configuration tables include but are not limited to the following:

- Details of communication between applications executed within the ARINC-653 RTOS
- ARINC-653 partition and module specification
- Allocation of applications on ARINC-653 partitions
- ARINC-664 switch configuration

The IMA Management team takes responsibility for the following tasks:

- Allocate IMA platform resources
- Manage the IMA platform configuration tables
- Perform verification testing on the integrated platform
- Coordinate qualification efforts on the module configuration

In order to perform verification testing and generate qualification evidence, the IMA management team performs integration testing at multiple levels including:

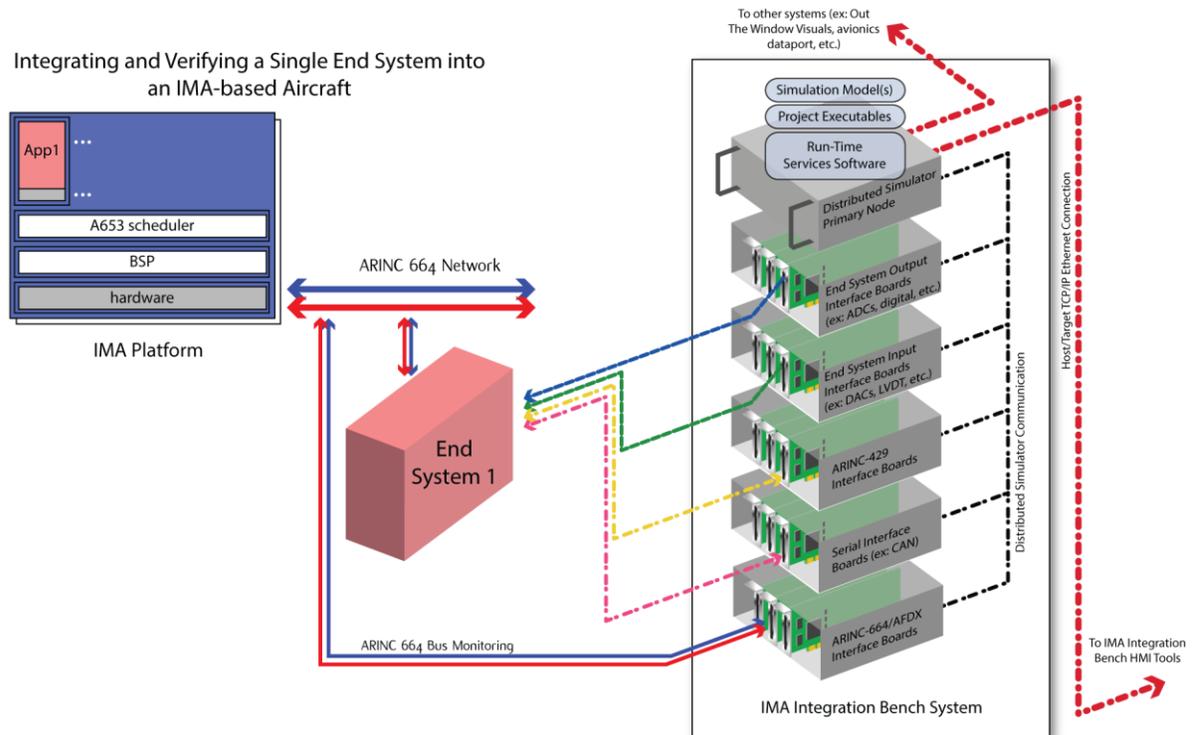
- Desktop software testing of the application
- Application testing on a representative hardware platform
- Functional verification on the real IMA platform
- Integration of all applications within an IMA integration facility

The task of functional verification of each aircraft system in the IMA Platform, first stand-alone, and later the integration and verification of all aircraft systems demands a new category of hardware-in-the-loop simulation testing facilities: ***The IMA Integration Bench***.

Where an Avionic Integration Facility will include a full flight deck, the full avionics suite, many of the aircraft electrical loads, and nearly every electronic system found in the aircraft, the IMA Integration Bench is limited to the IMA Platform, the ARINC-664 network and switches, and the electronic units connected to the ARINC-664 network; the remaining systems will be simulated in real-time.

## Integrating an Application and its End System

An aircraft system, supplied on an IMA-based aircraft, is initially developed by the supplier in a stand-alone manner. The supplier will develop the End System and its IMA application in the absence of the real IMA Platform computer system. After completing initial development, verification, and certification, the End System and its IMA application get integrated into the real IMA environment. The IMA environment includes the IMA Platform where the application is executed, the ARINC-664 network across which the application and End System communicate, and the intelligent IMA switches.

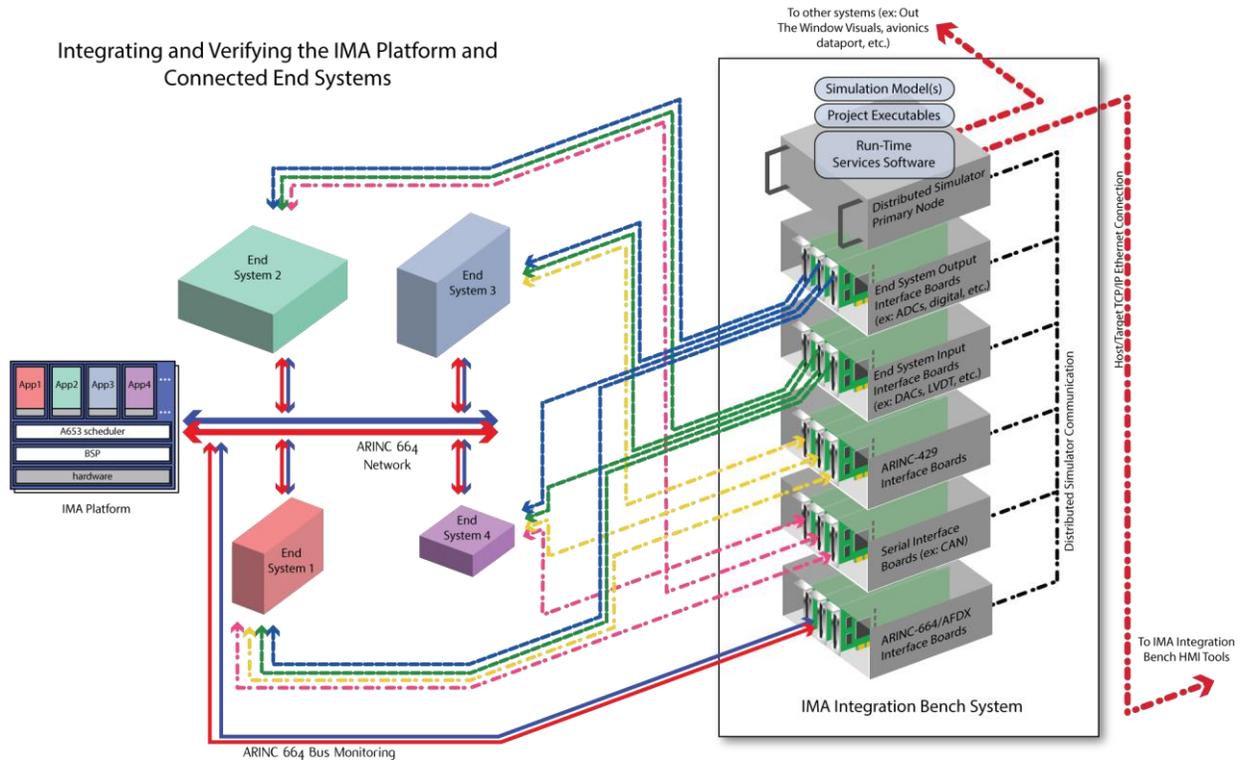


This stand-alone integration and verification task ensures that once operating within the real IMA environment, the aircraft system continues to operate (in a stand-alone manner) as expected (i.e. behavior matches system requirements and design).

This integration and verification effort is virtually impossible to accomplish without using hardware-in-the-loop (HIL) simulation. HIL puts the End System in a closed-loop simulation of those components too expensive to bring into the lab, e.g. landing gear, doors, hydraulic system, engines, generators, electrical system, etc. The End System communicates across the ARINC-664 network and is controlled by its application running on the IMA Platform. This simulation-based integration and verification effort includes exercising the system through normal operating and failure-mode conditions. Execution of these tests produces a significant amount of data. Later analysis and processing of this data generates evidence of the verification effort. This data is submitted as part of the IMA system airworthiness certification.

After integrating and verifying each aircraft system into the IMA environment in stand-alone operation, the next task is to integrate all systems together and verify the operation of the IMA resources when multiple applications are operating in normal and failure-mode conditions. The figure below illustrates

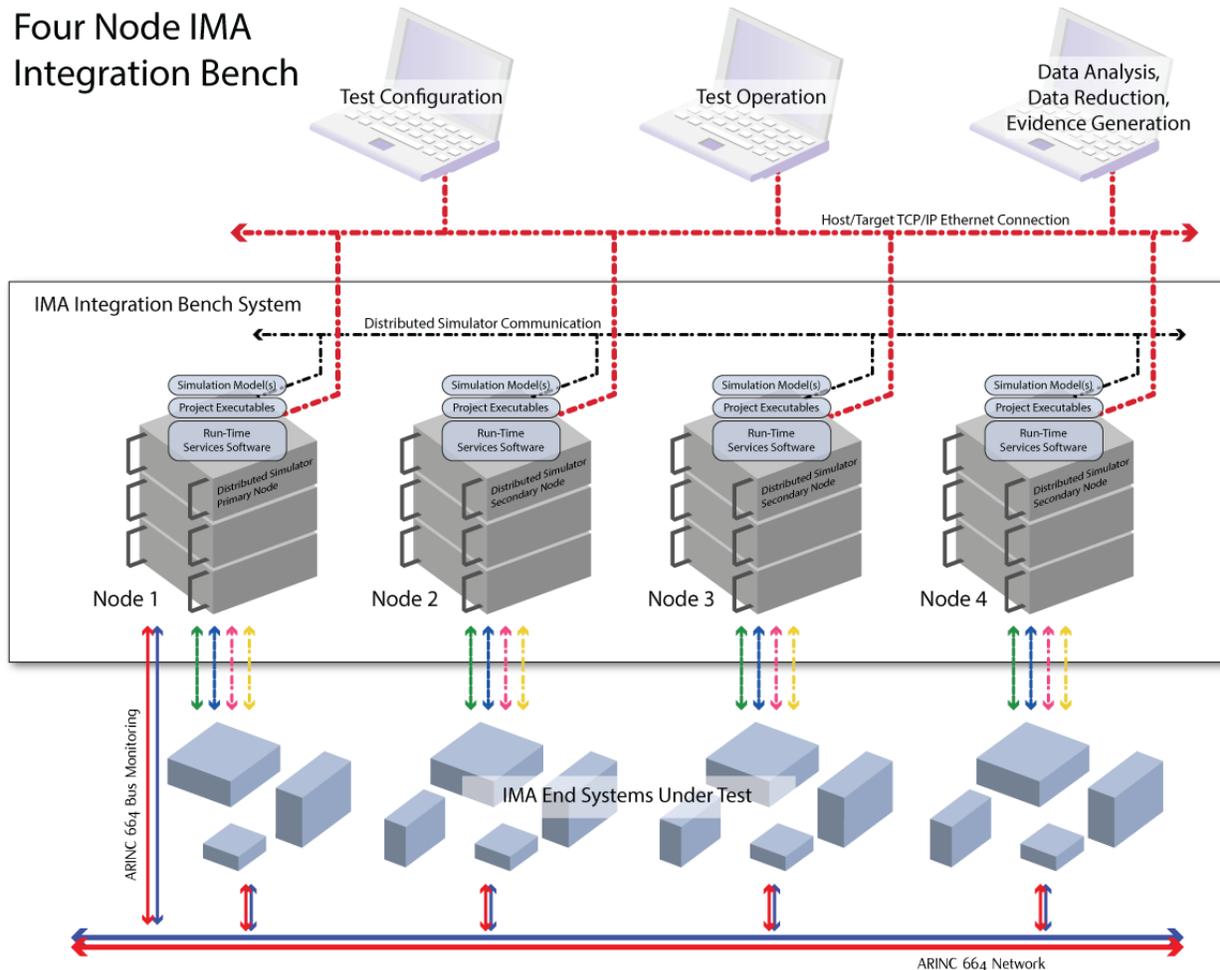
four aircraft systems integrated into the IMA environment, each with an End System and an application running on the IMA Platform. An IMA Integration Bench provides the hardware-in-the-loop simulation and verification facility to perform this important task.



## The IMA Integration Bench

Applied Dynamics works closely with airframers, avionics suppliers, and aircraft subsystem suppliers to support the adoption of IMA technology and to supply IMA Integration Bench systems used to integrate, verify, and certify IMA aircraft. The figure below illustrates a Four Node IMA Integration Bench.

### Four Node IMA Integration Bench



The IMA Integration Bench is a high-performance, multi-node, real-time simulation system with signal interfacing, cable interconnection, signal conditioning and electrical loads, programmable 28VDC aircraft power supply, remote-controlled aircraft power distribution, fault insertion, and a wide range of software features as required to perform interactive and automated integration and verification testing.

### IMA Integration Bench Signal Interface

The signal interfaces included in the IMA Integration bench put each IMA End System under-test into closed-loop operation with simulation models and/or other End Systems providing realistic and accurate behavior. The IMA Integration Bench interface signals can be divided into four main types. The table below lists these interface signal types.

Signal Type	Example
Sensor Emulation	Thermocouple, Resistive Temperature Device, Linear Variable Displacement Transducer (LVDT), Digital Encoder, etc.
Actuator Measurement	Torque Motor, Solenoid, Igniter, etc.
Serial and Databus	ARINC-664, ARINC-429, CAN, RS232/422/485, MIL-STD-1553, IEEE1394, etc.
Data Acquisition Channels	Analog, Digital

### Sensor Emulation

Sensor emulation sends analog or digital signals to the End System under-test, providing it with an accurate electrical emulation of the real sensors it expects to monitor as part of its system function. The value of these signals may be driven, in real-time, by the output of a simulation model or may be controlled directly from the test interface. There are a wide range of analog and digital signal types commonly found in a typical IMA Integration Bench.

### Actuator Measurement

Actuator signals are normally driven by an End System. For example, the FADEC controls a range of actuators to control fuel flow, to ignite combustion, to actuate the Variable Stator Vanes (VSV), to control bleed air, etc. In order to establish closed-loop HIL simulation, the real-time simulation computer must read the value of these actuation signals (often a measurement of electrical current) and use them as inputs to one or more simulation models. It is typically required to have the real-time simulation system include representative loads e.g. resistive, inductive, etc. The End System under-test will drive these loads and measure the characteristics of the total circuit. The circuit is measured by the End System to check for device or signal path failures. Therefore, the accuracy of the loads can be very important.

### Serial and Databus

Serial and databus signal interfaces are used to: Obtain values acting as inputs to one or more simulation models; simulate a complete End System and send necessary information to one or more real End System under-test; monitor serial and databus messages as part of a test case; interrupt and inject errors into serial and databus traffic. A range of different aircraft serial and databus communication types and quantities make up the complete aircraft communications network. Managing the network configuration through the course of the aircraft program, assigning configurations to interface channels for a given test configuration, connecting network signals to simulation model inputs and outputs, configuring network error injection, and configuring network monitoring for data acquisition are time consuming tasks. The *Working with Databus and Serial Interfaces* section of this whitepaper discusses methods and tools available to minimize this effort.

### Data Acquisition Channels

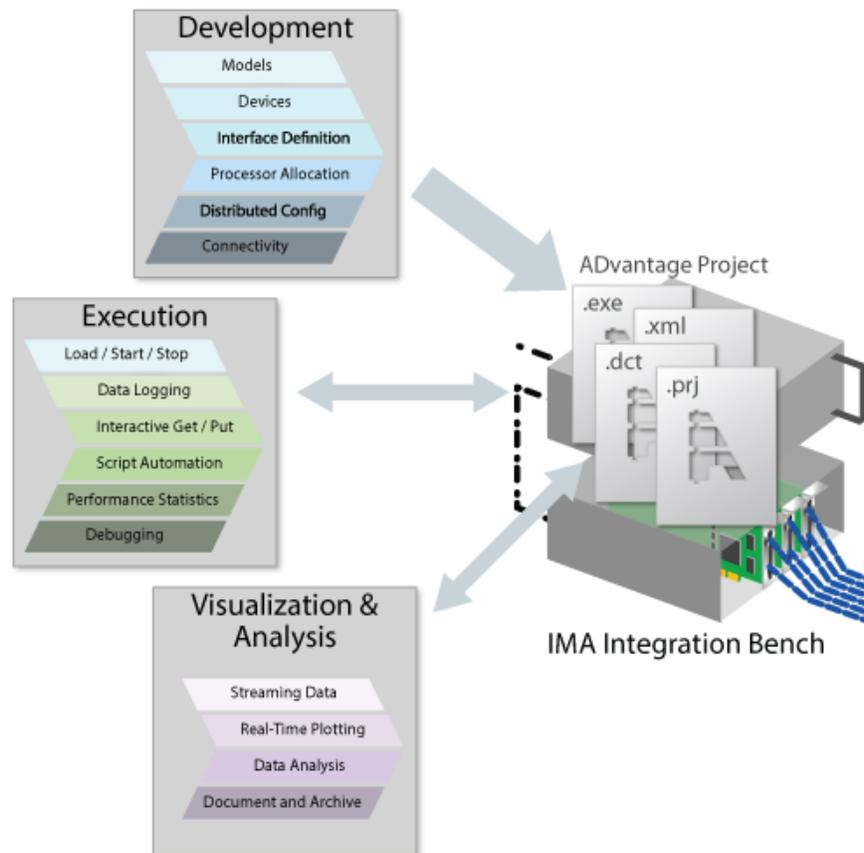
A well-designed IMA Integration Bench allows all of the sensor emulation, actuator measurement, and serial/databus signals to be time-stamp measured in real-time to generate the data required for analysis and evidence generation. Numerous auxiliary analog and digital signal channels (not required to emulate sensors or measure actuator signals as part of establishing closed-loop operation) are often required to make measurements of signals passing between two or more End Systems as part of one or more test cases.

## Working with the IMA Integration Bench

There are three main tasks performed using the IMA Integration Bench:

1. Development and configuration
2. System operation and test execution
3. Test visualization and analysis.

Applied Dynamics offers the ADvantage Framework - a total solution software platform to handle all three of these tasks and more.



### Development and Configuration

The development and configuration task involves preparing the IMA Integration Bench to execute one or more test cases. Over the course of an aircraft program, different End Systems will become available at different times and early in the program not all End Systems will be available at once. Rather than wait until all End Systems are available, the IMA Integration Bench is configured to test a subset of the aircraft systems. This will result in multiple projects; each with a different set of real systems and simulation models.

The task of development and configuration involves selecting the signal interface channels to be used, selecting the simulation models (and/or production code applications) to be executed, allocating

simulation models to real-time simulation computer processor cores, defining and documenting interface documentation (ICDs), configuring serial and databus interface channels, and establishing connections between simulation model inputs/outputs and signal interface channels.

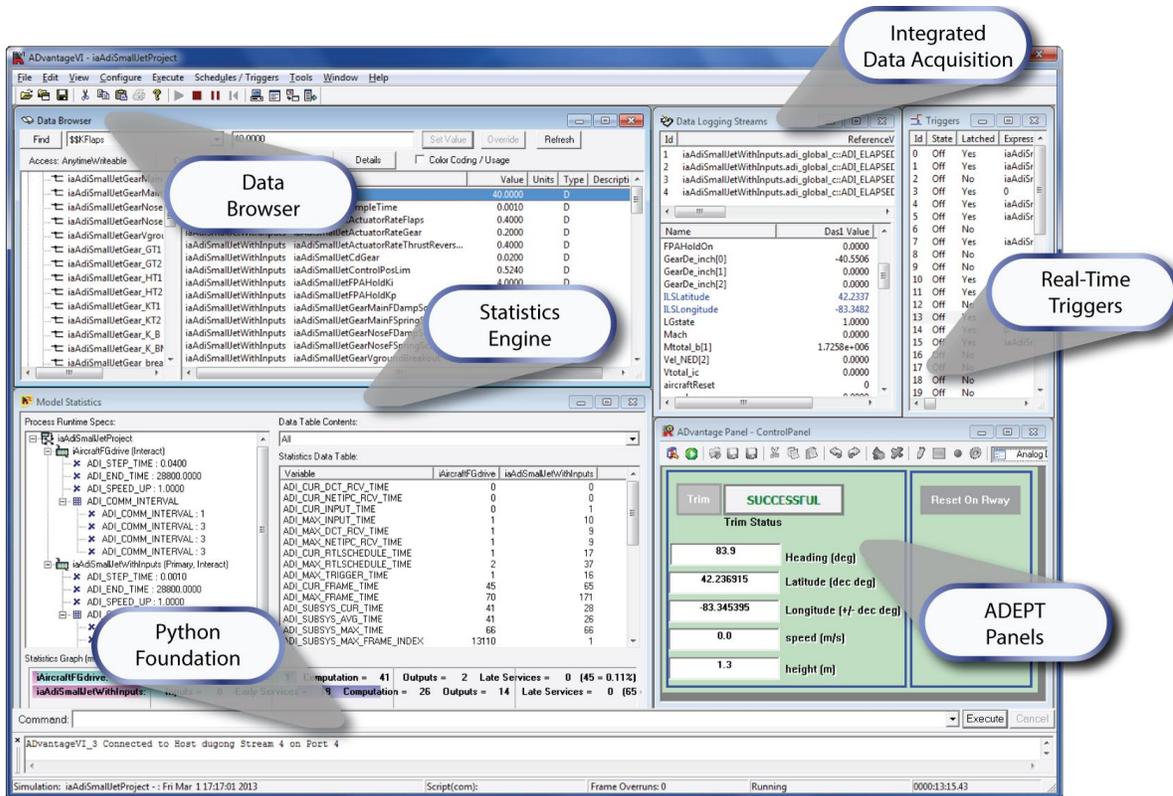
The ADvantage Framework includes a graphical development environment, ADvantageDE, where this information may be conveniently defined. Upon defining the configuration, the user presses the build button to generate the binary executables and configuration for execution on the real-time simulation computers. ADvantageDE makes heavy use of the GNU tools and uses a graphical definition of the project configuration to generate makefiles and other inputs fed into the GNU compiler collection tools, which build the binary executable files.

The ADvantage Framework supports a range of simulation model formats including Simulink, AMESIM, C, C++, and FORTRAN. Signal interface channels may be included using Commercial-Off-The-Shelf (COTS) computer boards from any board vendor who provides register-level programming documentation or source code. ADvantage includes support for hundreds of COTS I/O boards and can add support for additional boards upon request. Alternatively, a Driver Development Kit (DDK) is available for users who want to develop their own device drivers. Most users are able to configure and develop their projects with drag and drop and get a system up and running in an hour or less, depending on the complexity of their first project.

### Execution and Operation

The task of executing and operating the IMA Integration Bench involves loading the appropriate, previously developed project; initializing models, e.g. trim the aircraft model; specifying which data to acquire for a given test; starting real-time execution for the total HIL simulation; executing interactive and/or automated test cases; archiving logged data; and, reporting any observed faults, problems, or whether the test case ran without issue.

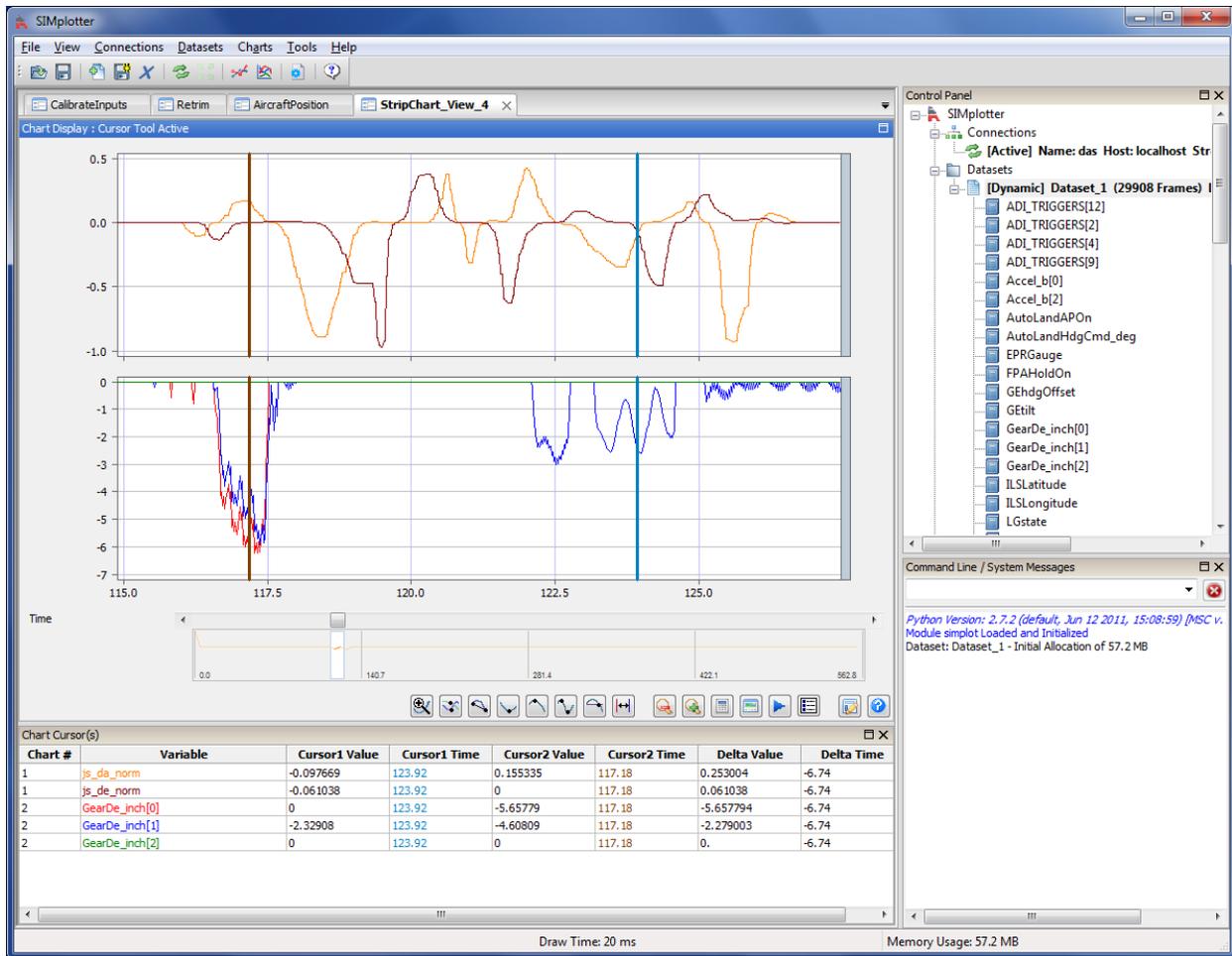
The ADvantage Framework includes a graphical operator environment, ADvantageVI. ADvantageVI provides a complete set of capabilities for interactive and automated test case operation including: a Data Browser, data acquisition interface, real-time scripting interface, Python scripting editor, Python command line, real-time Statistics View, Device Status Browser, graphical Test Automation Toolbox, graphical panel toolbox, and much more. ADvantageVI is built on a foundation of Python scripting technology which provides modular, object-oriented test scripting and automation, and enables the ADvantageVI application to be extended with hundreds of open source Python libraries. The figure below illustrates the ADvantageVI operator application.



### Visualization and Analysis

Visualization and analysis of the IMA Integration Bench test cases involves presenting data in a convenient format to support the identification or absence of errors. This may involve an operator monitoring test case progress while the test is running or post-run efforts. The IMA Integration Bench is a valuable testing asset and therefore, for the sake of efficiency, it is common to run tests in batch mode, log data, then have users and analysts visualize and analyze the generated data away from the IMA Integration Bench as a post-run effort. The purpose of the visualization and analysis task is to run interactive and/or automated methods to analyze, reduce, and report on the identification or absence of errors.

The ADvantage Framework includes the popular Python-based SIMplotter tool for visualization and analysis. SIMplotter supports real-time, TCP/IP Ethernet streaming XDR data allowing real-time and offline post-run visualization and analysis. SIMplotter also supports NumPy and SciPy numeric and scientific methods libraries allowing highly sophisticated visualization and analysis methods.



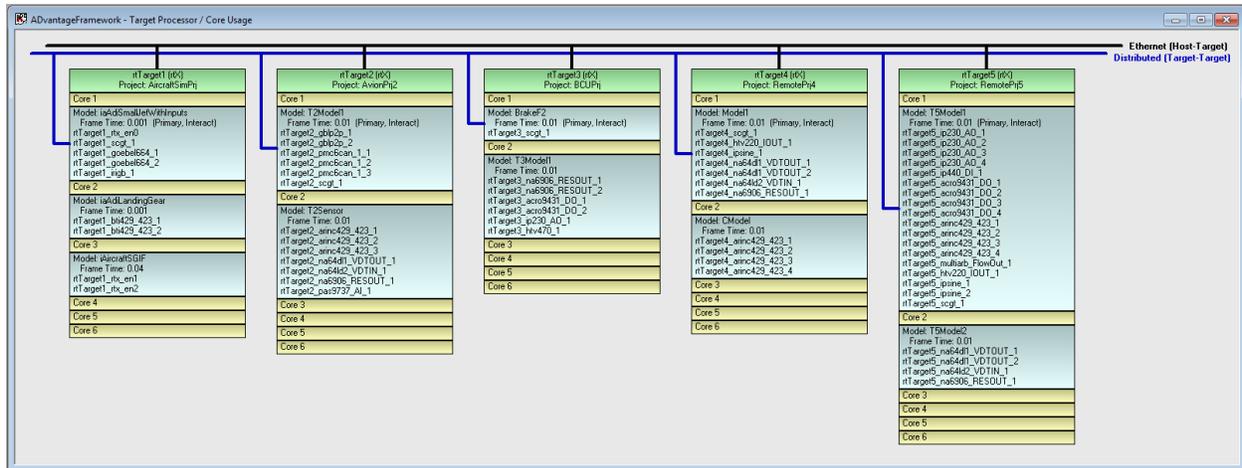
SIMplotter may be launched on any PC, attached to a data stream, and work with live, real-time data. The figure above shows the SIMplotter application.

### Distributed System Architecture

Performing HIL simulation with dozens of IMA End Systems in the loop can require thousands of signal interface channels, hundreds of serial and databus channels, and dozens of simulation models. This results in a tremendous computational power requirement. Through more than a decade of experience working with high-density, high-fidelity HIL simulation, Applied Dynamics has determined that an IMA Integration Bench with a distributed architecture offers the best-performing solution. It is common to use four or five high-performance, 6-core or 8-core PC-based real-time simulation computers connected via ultra-high-speed distributed communication bus in order to meet the computational needs of an IMA Integration Bench.

The ADvantage Framework makes distributed real-time simulation painless with its ADvNET Toolbox. ADvNET configures and operates a multi-node, multi-core, distributed HIL simulation project as though it were a single real-time computer. Node-to-node communication is handled seamlessly without the

need to configure device drivers or network interfaces. Simulation models and I/O device handling are easily assigned to processor cores on the real-time simulation nodes. The figure below shows the Target Processor / Core Usage view within ADvantageDE showing the assignment of models and devices in a five node real-time simulation project. In this example, the first real-time simulation computer node is using three of its 8 processor cores. The first core is executing a Simulink aircraft simulation model as well as handling computation associated with some ARINC-664 interface channels, an IRIG-B distributed clock, and some other interface channels. The second processor core is executing a Simulink landing gear model and handling ARINC-429 computational tasks. The third core is driving two Ethernet interface boards used to send and receive data for out-the-window scene generation. The remaining four real-time computer nodes are using two of their six processor cores for a range of tasks. The unused processor cores provide additional computational power used with more computationally intensive test configurations.



## Data Dictionaries

The interface specification for each simulation model and the signal interface specification for each End System under-test are a critical component of each IMA Integration Bench project. These interface specifications are used to connect the pieces during the development and configuration effort. Later, these interface specifications represent the test interface specification used to set values, get values, configure data acquisition, and write automated test case scripts.

The ADvantage Framework includes the concept of a data dictionary. Each data dictionary provides the interface specification to a project assembly (a simulation model and the collection of I/O assigned to a processor core). ADvantageVI includes a Data Browser that allows the user to conveniently browse the data in each project assembly. A given data dictionary item may be dragged from the Data Browser to the Data Logging window to easily configure data acquisition or dragged to the command line to facilitate the execution of command line functions. The ADvantage Framework also includes automatic generation of data dictionaries for Simulink models as well as C models.

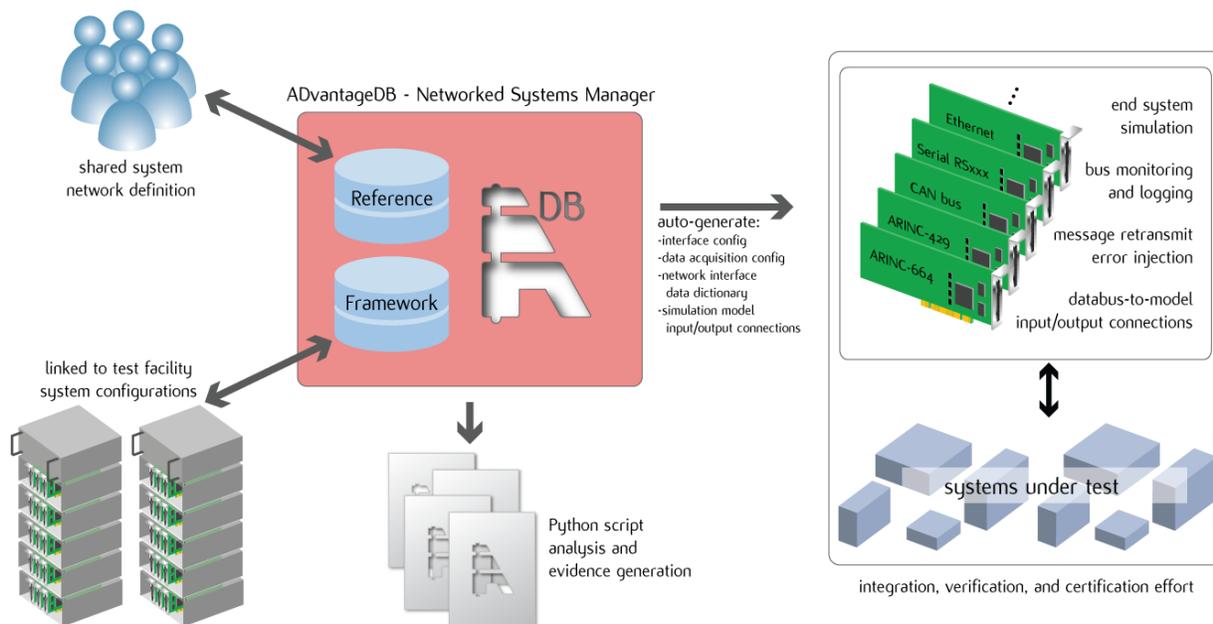
## Data Acquisition

Data acquisition during IMA Integration Bench test execution is key. As a test case is executed, data will be collected to record the test case stimulus and the effect this stimulus has on signals of importance. The data collected will typically vary from test to test and reconfiguration must be quick and easy.

The ADvantage Framework includes a powerful data acquisition system (DAS). The DAS includes run-time services executing on the real-time computer, streaming data across TCP/IP Ethernet; a desktop server to collect the real-time streaming data and make it available to multiple desktop clients; support for the configuration and receipt of streaming data from the ADvantageVI and SIMplotter applications; and open source libraries used to interface the DAS data streams with user applications (as a data source or a data recipient). The DAS uses the well-known XDR compact binary data format, originally developed by Sun Microsystems, which offers optimized distributed data streaming performance.

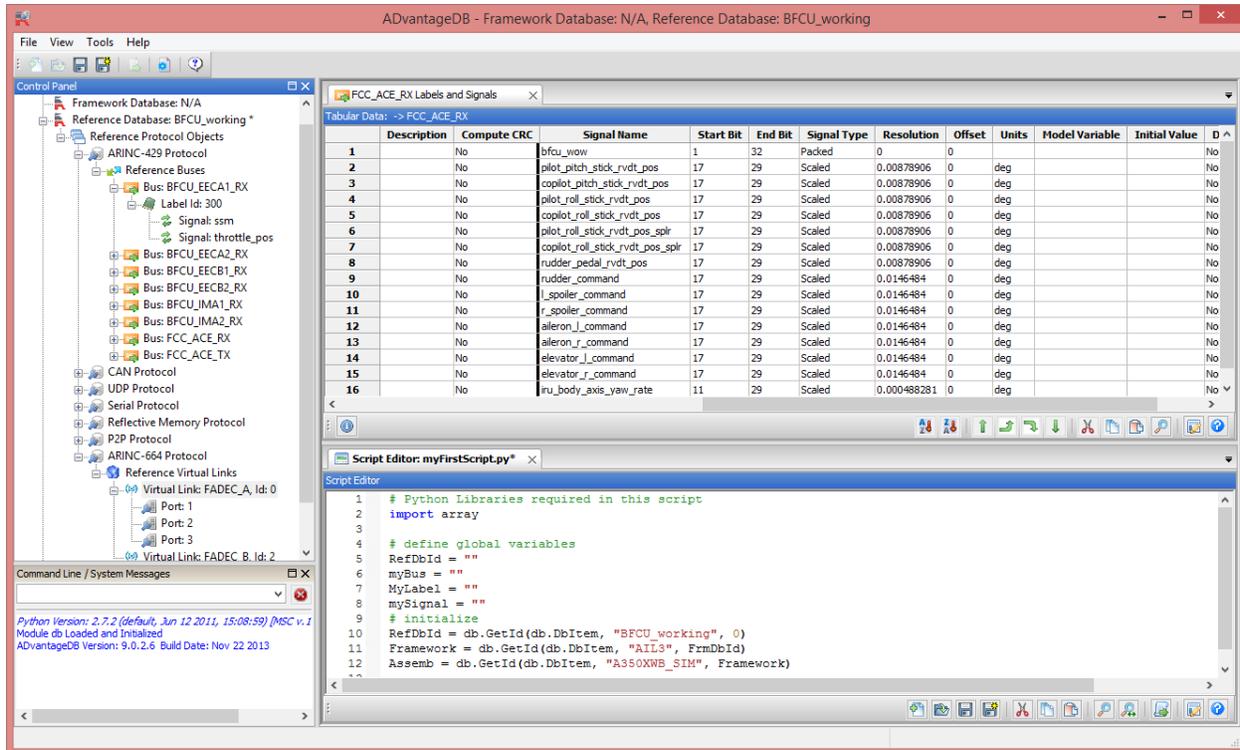
## Working with Databus and Serial Interfaces

A challenge of the IMA Integration testing task is working with the dozens and even hundreds of serial and databus network communication channels. This involves correctly defining the configuration (i.e. message packing, Virtual Links, BAG, SDI, SSM, etc.) of each network channel across a range of network types including ARINC-429, ARINC-664, CAN, RS-485, UDP Ethernet, and more. Network channel definitions assigned to serial and databus interface boards configure each interface channel within the real-time computer nodes. Signals moving across this dense network must be connected as inputs to simulation models; and outputs from simulation models need to be sent out on the network; all in a real-time deterministic manner. Furthermore, these network channel definitions will typically change over the course of an aircraft development program as the design of each End System evolves.



The ADvantage Framework includes the ADvantageDB tool designed specifically for defining and working with aircraft network communication. ADvantageDB configures the test facility for a given set of test cases with minimized effort. ADvantageDB uses standard XML data file format preferred for

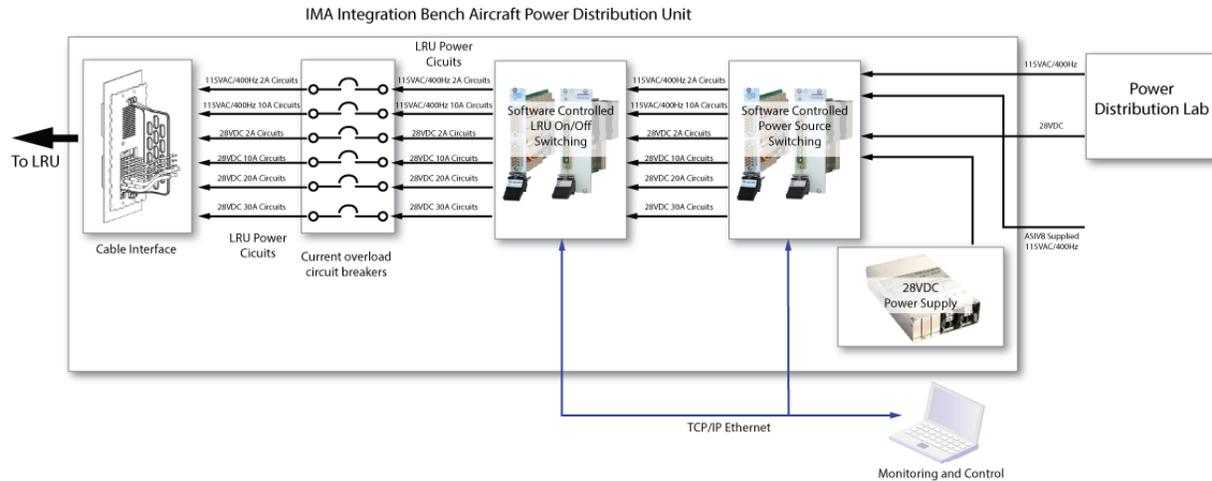
revision control and change tracking and includes a Python scripting foundation for performing analysis and automating tasks. The figure below shows the ADvantageDB tool.



ADvantageDB includes the concept of a Reference Database representing the golden standard definition of the complete aircraft network. Reference Objects (bus definitions) are applied to a Framework Database to assign bus definitions to bus interface channels associated with a given IMA Integration Bench project configuration.

### Power Distribution

Another capability commonly included within an IMA Integration Bench is an Aircraft Power Distribution Unit (PDU). The PDU provides software-controlled 28VDC and/or 115VAC/400Hz circuits used to connect to the End Systems under-test. Power to each aircraft LRU can be software controlled as part of the test case. Script-automated test cases will often require that power to a given End System be disconnected, as seen with failure mode conditions in the aircraft power generation and distribution systems.



Other features of the PDU include: current-overload circuit breakers to avoid damaging equipment e.g. faulty prototype End System; software controlled power source switching used to connect the IMA Integration Bench to other test labs; an assortment of different circuit amperage levels. The figure above illustrates an IMA Integration Bench PDU.

With ADvantage based IMA Integration Bench facilities, the PDU may be added to the ADvantageDE project as a distributed node, providing clock synchronization and a common interface for user interaction and test case scripting.

### Modular Test Language

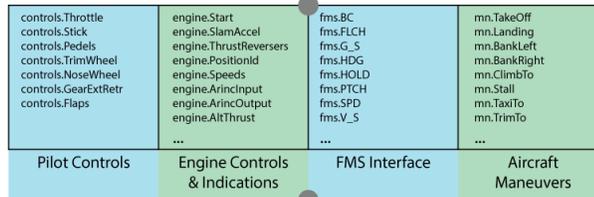
Operating the IMA Integration Bench involves executing a collection of test cases to take the IMA End Systems under-test through a range of realistic flight conditions. The ADvantage Framework provides Python scripting interfaces and integrated support for Python editing and execution within ADvantageVI and SIMplotter. The core test language scripting interface includes all the required functionality for test scripting including: project and application methods; real-time target interaction methods; data acquisition methods; real-time scripting methods; test user interface methods, and plotting and analysis methods.

The core ADvantage test language methods provide comprehensive low-level control over the IMA Integration Bench system. The best-in-class approach to work with the IMA Integration Bench is to develop a modular test language on top of the base Python API that includes a set of higher-level aircraft interface functions. These aircraft specific functions combine awareness of the aircraft interfaces (through the data dictionaries) and knowledge of aircraft operation to build a set of functions used for day-to-day control of the simulation based facility. Types of aircraft interface functions may include: pilot controls; engine controls & indications; Flight Management System (FMS) interface; standard aircraft maneuver; etc.

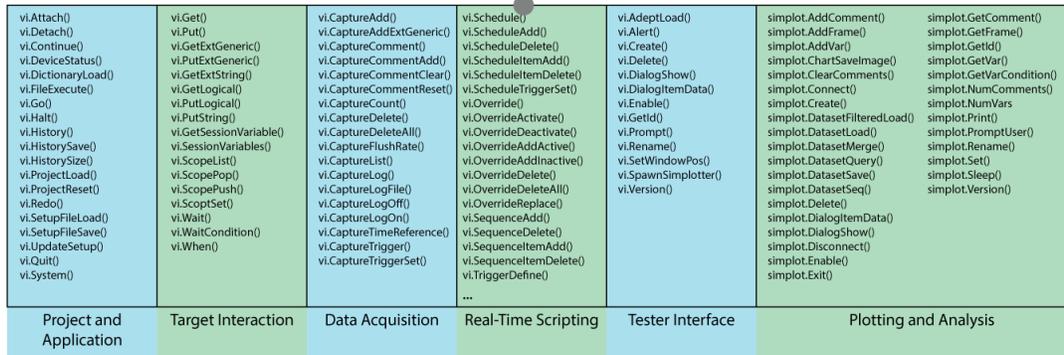
Aircraft Integration Test Cases

- takeoffAndTakeoffSpeedsTest
- accelerateStopDistanceTest
- takeoffPathTest
- climbTest
- normalLandingTest
- controlabilityAndManeuveringTest
- LongitudinalControlTest
- DirectionalAndLateralControlTest
- MinimumControlSpeedTest
- TrimTest
- StaticStabilityTest
- DynamicStabilityTest
- StallTest
- HighSpeedCharTest
- OutOfTrimCharTest
- EngineControlAndIndicationsTest
- LandingGearControlAndIndicationsTest
- AircraftElectricalTest
- ...

Aircraft Interface Functions



ADvantage Core Test Language Methods



These higher-level, aircraft-specific functions then form the set of test language commands used to write the aircraft integration test cases. In addition to basic functionality, the aircraft interface functions may also support the following:

- Version traceability information
- Author, test, user reporting
- Graphical user interface reporting
- Automatic report generation
- Generic interface to data dictionaries to support a range of different aircraft

The figure above illustrates the three layers associated with a modular test language.

## Summary

Integrated Modular Avionics (IMA) represents a revolutionary change in commercial and military aircraft technology. The Airbus A380 and the Boeing 787 are the two highly advanced aircraft programs that led the commercial push to embrace an IMA architecture. The commercial and military aircraft industry is not usually characterized by sweeping, wide-reaching technology change but rather by small incremental change. So, the many aspects of new technology insertion associated with IMA represent something significant.

IMA is a departure from the 'federated avionics' architecture. The IMA architecture uses a distributed architecture where a given avionics function, e.g. landing gear extension and retraction, is split into the software application and the End System. The trend in aircraft systems shows a steady increase in the number of sensors and electrical loads. The weight of cabling translates directly into increased aircraft weight and fuel consumption. The IMA architecture offers a path to reduce the operating cost of a given aircraft and build a more compelling product.

With aircraft systems' increasing dependence on software and electronics, and on data sharing, the challenge of integrating these systems and verifying system interoperability has increased. In an IMA-based aircraft this task is suddenly far more challenging. Functional verification of each aircraft system into the IMA Platform, in a stand-alone manner, and later the integration and verification of all aircraft systems demands a new category of hardware-in-the-loop simulation testing facilities: The IMA Integration Bench.

Important aspects of the IMA Integration Bench include:

- Distributed system architecture with multiple nodes and multi-core processors
- Data dictionary interface to simulation models, signal interfaces, and aircraft network traffic providing crucial Interface Control Documentation (ICD)
- Flexible, reconfigurable acquisition of data as required for a given test case
- Flexible configuration of aircraft network channels, e.g. ARINC-664, ARINC-429, RS-485, etc., and their assignment to network testing interfaces
- Software controllable aircraft power distribution
- A modular test language that includes traceability, revision control, user interaction, automatic report generation, and the ability to abstract and inherit for multiple aircraft programs

Applied Dynamics works closely with airframers, avionics suppliers, and aircraft subsystem suppliers to support the adoption of IMA technology and to supply IMA Integration Bench systems used to integrate, verify, and certify IMA aircraft. To learn more about how a technology partnership with Applied Dynamics can drive success with your organization's adoption of IMA technology, please contact us and speak with an application engineer.

CONTACT APPLIED DYNAMICS

 USA

Applied Dynamics International, Inc.  
3800 Stone School Road  
Ann Arbor, MI 48108-2499  
USA  
Tel: 734.973.1300  
Fax: 734.668.0012  
Email: [adinfo@adi.com](mailto:adinfo@adi.com)

 USA

Mr. Alan Strech  
Applied Dynamics International, Inc.  
32201 Crystalaire Drive  
Llano, CA 93544-1240  
Ph: (661) 944-1969  
Email: [alan@adi.com](mailto:alan@adi.com)

 CHINA (PRC)

Beijing  
Ensky Technology Co., Ltd.  
Mr. Wang Liwu  
Room 1031, Section C, Chaowai SOHO  
No.6 B, Chaowai Street  
Chaoyang District, Beijing  
100020, China  
Tel: 010-59009377, 59009378;  
Mobile: 13601227682  
Fax: 010-59002883  
Email: [wang.liwu@ensky.com.cn](mailto:wang.liwu@ensky.com.cn)

 JAPAN

Kyokuto Boeki Kaisha Ltd.  
Mr. Yuichi Wada  
New Otemachi Bldg., 2-1, Otemachi 2-Chome,  
Chiyoda-ku, Tokyo 100-0004  
Japan  
Tel: 81.3.3244.3823  
Fax: 81.3.3246.2765  
Email: [wada@kbk.co.jp](mailto:wada@kbk.co.jp)

 UNITED KINGDOM

Applied Dynamics International, Ltd.  
Ms. Debbie Beech  
No. 1 Mill  
The Wharf  
Shardlow  
Derbyshire DE72 2GH UK  
Tel: 44.0.1536.410077  
Email: [adiukinfo@adi.com](mailto:adiukinfo@adi.com)

 FRANCE

ASC – Applied Software & Consulting  
M. Gilles Derio  
112, av Kleber  
75784 Paris Cedex 16  
France  
Tel: 33.0.1.47.55.74.00  
Fax: 33.0.1.64.22.94.13  
Email: [gilles.derio@asc-sas.com](mailto:gilles.derio@asc-sas.com)

 ITALY

Otopos  
Sig. Gianfranco Cattadori  
Via Tetto Nuovo 10/G  
10025 Pino Torinese, Torino Italy  
Tel: 39.011.811.1145  
Mobile: 39.335.60 75 110  
Fax: 39.02.700446492  
Email: [g.cattadori@otopos.it](mailto:g.cattadori@otopos.it)

 KOREA

Anawell Corporation  
Ms. Y.E. Park  
Shinil Uto Vill Rm 711  
735-11 Yuksam-Dong, Kangnam-Ku  
Seoul 135-080  
Korea  
Tel: 82-2-554-2173,  
Fax: 82-2-554-2175  
Email: [yepark@anawell.com](mailto:yepark@anawell.com)