Managing Multi-protocol Network Definitions in Aircraft Simulation and Test

Brian J. Hale
Applied Dynamics International, Ann Arbor, MI, 48108

Modern aircraft have grown increasingly dependent on data networks to share data between subsystems. As these networks grow in complexity, managing the network definitions throughout the development and test processes has become a critical and time consuming task. This paper will discuss the evolution and advancement of software tools that have been created to address this crucial task in aircraft simulation and test laboratories.

I. Introduction

In an aircraft test environment, multiple communication protocols are required and utilized. These include but are not limited to ARINC 429, ARINC 664, CAN, UDP, and serial protocols. Messages must be defined and configured for each protocol. For a complex system, the number of signals and messages that must be defined can quickly spiral. The contents of the messages themselves can evolve as the system becomes increasingly mature. In a multi-node, distributed simulation test setup managing the network definitions can be a monumental task.

The ability to manage and modify the entire network definition from a single application allows for a significant increase in test efficiency and lab operability. Additional benefits include the capability to provide a snapshot of the aircraft network definition at any point of the testing process – crucial for any certification effort. Applied Dynamics International (ADI) has developed a new software application, ADvantageDB, that allows the entire multi-protocol definition to be captured, developed, traced, and shared.

ADvantageDB provides users the ability to build a reference database – a library of bus and interface definitions that can be used multiple times throughout a project and shared between test systems. This library of bus definitions contains all messages that a user may desire to transmit or receive throughout the aircraft. This master reference library can often be shared between aircraft simulation systems. Modifications to the reference database are automatically propagated to a framework database. The framework database is linked to hardware communicating with the system under test. This linked database architecture allows for rapid system reconfiguration. A reference bus may be used multiple times throughout a system. Users are also able to break the link between the databases, for simple one-off test cases and investigation. The flexibility offered by this architecture has proven to be an asset in iron bird and aircraft system integration laboratories.

The information contained within the database definition is used for a variety of purposes. Primarily, the network and bus definitions are used to generate supporting files that are used in high and low-level simulation software that allows the system to transmit and receive data during the simulation run. Additionally, the database information contained can easily be harnessed with Python scripting to be used for report generation and simulation load analysis.

II. Aircraft Test Lab Applications

Multiple real-time simulation labs are often utilized for integration and testing of the same aircraft. While these labs each have distinct capabilities, requirements, and testing purposes, there are areas of overlap where testing and workflow gains are to be had. One of the largest areas of overlap is a common aircraft network definition. Although each lab may not require the ability to communicate with and simulate every subsystem, the bus definitions that the labs have in common are often identical. As aircraft simulation has grown in scope and utility, several types of labs have developed as industry standards: avionics integration facilities, iron bird facilities, a system integration bench, and cockpit development and test facilities.

A. Avionics Integration Facility

The avionics integration test facility is used for testing all avionics and electrical subsystems on an aircraft. Throughout the test and development schedule, real aircraft hardware is brought in and out of the system, generally in the form of line-replaceable units (LRUs). In some cases, LRUs will not be available or desirable, and their functionality will be simulated entirely. Not surprisingly, the avionics integration facility will often have the largest network definition amongst the aircraft test labs. Considering all subsystems in the aircraft, a wide variety of protocols are used heavily. The ARINC 429 definition alone will easily contain thousands of signals. Additionally, CAN and UDP protocols are widely utilized.

B. Iron Bird Facility

The primary purpose of the iron bird test facility is to test the aircraft's flight control system with a full hydraulic system and real control surfaces integrated into the lab. In modern fly-by-wire aircraft, control surface actuators may be electromechanical or electro-hydrostatic. Similar to an avionics integration facility, the iron bird facility will generally have full cockpit and the avionics to support flight control. While the testing emphasis is not on avionics integration, there will be a large and diverse network definition nevertheless.

C. System Integration Bench

The system integration bench is a relatively new addition to the standard aircraft simulation labs. As Integrated Modular Avionics (IMA) have come into favor in recent years, testing requirements have necessitated a need for a dedicated lab. The system integration bench is used for this purpose. With an IMA architecture, various subsystem communication is handled with a dual-redundant ARINC 664 ethernet backbone. The subsystems do however utilize various other protocols, including ARINC 429 and CAN, before being converted to ARINC 664. Therefore a large, multi-protocol network definition still needs to be defined and maintained.

D. Cockpit Development and Test Facility

The cockpit test facility is used for flight deck development and human factors testing. The high fidelity visual system also allows these labs to be used for simulation control law verification, and pilot-in-the-loop testing. The network definitions for these types of facilities are generally on a smaller scale than their test lab counterparts, but ARINC 429 and UDP protocols are often utilized for communication with cockpit avionics and out-the-window display systems, respectively.

III. Creating and Maintaining an Inclusive Database Library

The original aircraft network definition is gathered together from a wide variety of sources. Various subsystem suppliers will provide interface control documents (ICDs) to the aircraft manufacturer containing in-depth details on the signal definitions for all necessary messages and labels. Python scripts can be developed by the aircraft manufacturer to import the ICDs, which are often provided in different formats, directly into ADvantageDB. As these ICDs get modified and updated during the design process, the scripts can be reused to easily import changes. In an ideal scenario, the subsystem suppliers themselves already define their system communication definitions with ADvantageDB, and any updates or modifications can simply be merged with the existing database.

There are two types of databases defined in ADvantageDB: a reference database and a framework database. Within each of these databases, messages are segregated by protocol. The reference database is a library of all messages that may be used for any particular simulation run. A reference database may be shared between aircraft test labs as well as network definition creators (subsystem suppliers). The framework database is an application of network definition objects on the physical hardware that is used during the simulation run. A reference database can optionally be linked to a framework database, such that any updates to the reference are automatically propagated to the framework database. Figure 1 below diagrams this relationship. Note that a framework database can be made up entirely of reference objects, a subset of reference objects, or be completely independent from the reference database.

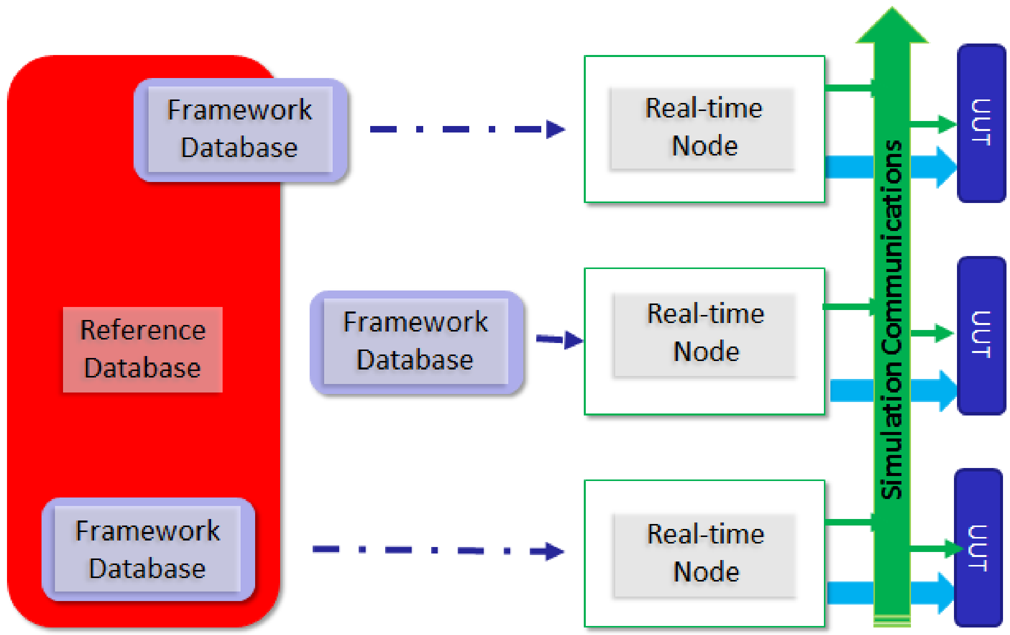

Figure 1. Reference and framework database relationship

For each protocol, the reference framework contains reference objects at a logical level. Generally, the reference object will be a message, interface, or bus definition. These reference objects are then assigned to the framework database. For a distributed system, they are assigned to the appropriate physical hardware target, model, and IO device for a given lab architecture and test case. Once defined in the framework, busses, messages, or hardware channels can be enabled or disabled to optimize testing. Figure 2 below shows an example framework database, along with the reference database containing supported protocols and their corresponding reference objects.

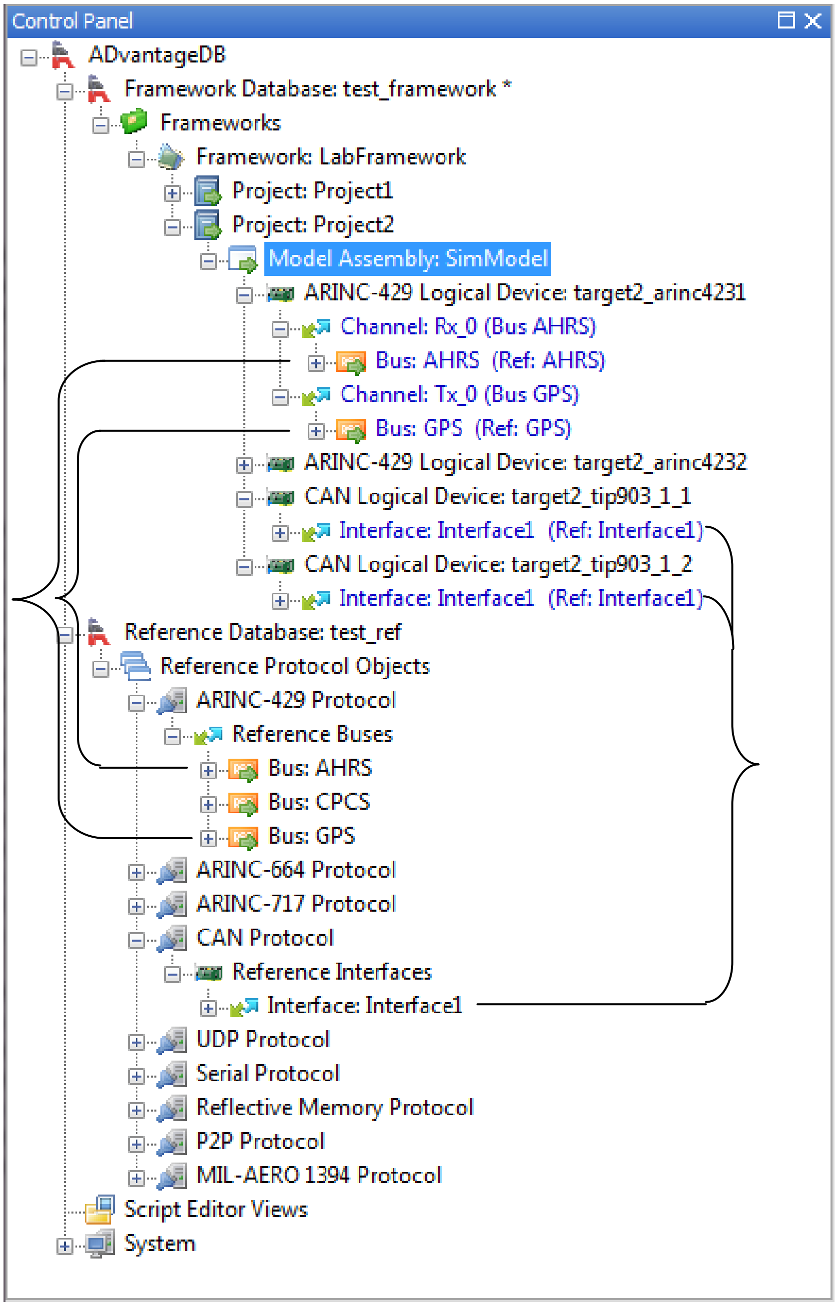

Figure 2. Example Simulation Framework Hierarchy

IV. Integration with Real-Time Simulation Systems

The database information stored within ADvantageDB can be used in a variety of ways. Currently, the primary function ADvantageDB provides is the ability to generate files that are used by communication board hardware drivers to give users the ability to transmit, receive and interact with simulation data throughout during the test run. Dictionary files generated are used by ADI's run-time tool, ADvantageVI for memory mapping and allocation. Schedule files are generated that allow users to drive or have their model driven by databus signals received by the hardware and processed in ADvantageVI. Figure 3 below displays the role of AdvantageDB within a real-time simulation system.

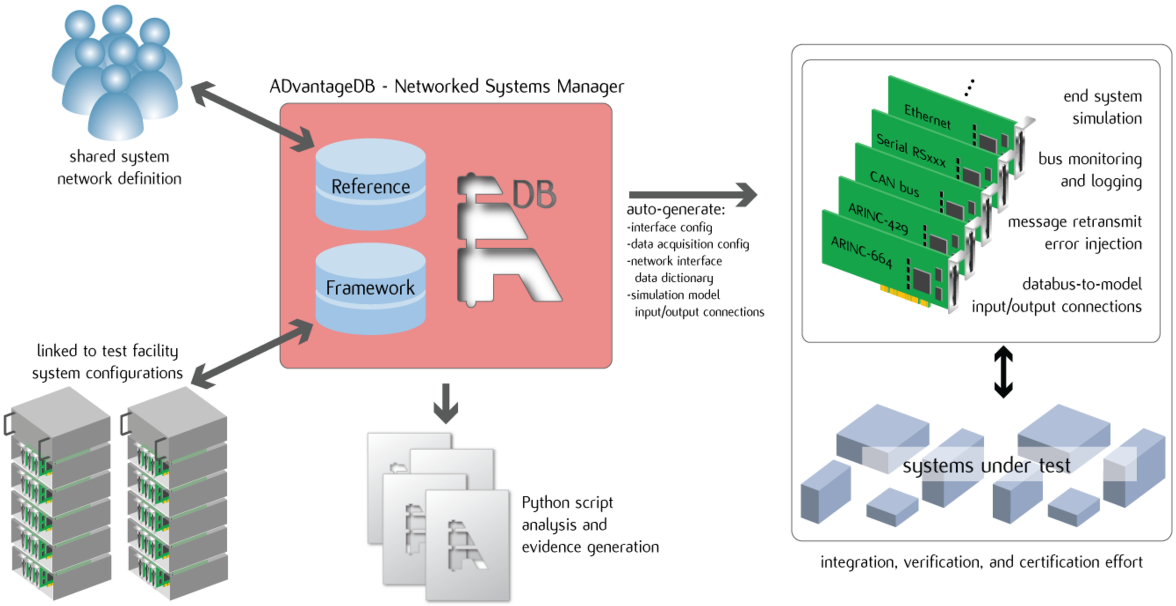

Figure 3. ADvantageDB's role within the simulation system

The network definition contained within ADvantageDB can also be used for purposes beyond direct integration with other ADI simulation tools. ADvantageDB includes a Python module with functions to extract information from the network definition at any level. This ultimately allows the user to use and then format this information for other systems, or for report generation and supporting documentation.

V. Conclusion

The ability to manage and modify the entire aircraft network definition from a single application allows for a significant increase in test efficiency and lab operability. Additional benefits include the capability to provide a snapshot of the aircraft network definition at any point of the testing process – crucial for any certification effort. ADvantageDB stands as the only tool that supports combined protocols allowing the entire aircraft network definition to be captured, modified, and shared in one source. As new communications protocols come into favor by aircraft manufacturers, the ADvantageDB tool is poised to expand and support these protocols on a common platform.

References

Savaglio, C. "Development of a Real-Time Capable Integrated Aircraft Model for Test, Integration, and Development Support", *2013 Aviation Technology, Integration, and Operations Conference,* AIAA Aviation, (AIAA 2013-4271).

[1]James, S., Savaglio, C., "The Expanded Reach of Simulation Based Aircraft System Verification and its Software Capability Requirements", WWW.ADI.COM, Applied Dynamics International, Ann Arbor, MI, (unpublished).