The Global Leader in DDS

June 2011

Flexible Data-Centric UAV Platform Eases Mission Adaptation

Edwin de Jong, Ph.D.

US HEADQUARTERS
Real-Time Innovations, Inc.
385 Moffett Park Drive
Sunnyvale, CA 94089
Tel: +1 (408) 990-7400
Fax: +1 (408) 990-7402
info@rti.com

Executive Summary

Because Unmanned Air Vehicles (UAVs) are invariably designed for specific missions, they tend to be difficult to adapt to changes in mission once built and deployed. Building a UAV that is readily adaptable to different mission parameters requires a new platform approach to UAV architecture. The biggest advantages are significant savings in cost, complexity, and time in the operational phase. Part of the technical challenge in building a UAV platform is providing an integration framework that supports this approach. Such an integration framework can be implemented through the use of middleware to abstract the details of point-to-point data transmission away from the individual components of the UAV. This type of middleware has been created as the Data Distribution Service for Real-Time Systems (DDS), a standard published by the Object Management Group (OMG) and currently used in implementations of both UAVs and ground stations.

Introduction

The growing drive to employ UAVs for a variety of defense and commercial purposes has exposed weaknesses and limitations of traditional design strategies. Because UAVs were invariably designed for specific missions, they tend to be difficult to adapt to changes in mission once built and deployed. A new set of mission profiles necessitates the design of yet another UAV. Consequently, over time, manufacturers and user organizations have to maintain production and support lines for multiple UAV systems, all having a common architecture but needing different performance, connectivity, or data characteristics.

Ideally, two or three UAV designs could serve as platforms that could be readily adapted to a variety of different missions. The platforms would be differentiated by their physical characteristics, such as power plant, airframe, and fuel capacity, with complementary hardware and software. For mission flexibility, components should be field replaceable and hot swappable on the platform without disruption. The software infrastructure should provide automatic discovery and configuration, so that components can be added and removed dynamically on a live system and immediately recognized and integrated with the system. Also, nodes should be able to restart after failure at any time, and applications should be able to start in any order.

Conceptually this idea is easy to understand, but technically it cannot be easily implemented using current design principles. Today, each vehicle design requires a special-purpose ground station, which is well integrated but expensive. UAV flight software is typically redesigned for each new vehicle, and even sometimes for different missions or payloads on the same vehicle. Building a UAV that is readily adaptable to different mission parameters doesn't seem feasible.

Rather, it requires a new architecture for systems design and implementation - one that provides both the features and the flexibility to take on multiple roles and missions without returning to the drawing board for substantial modifications. Consider this the "platform approach" to UAV design. By itself, the UAV platform isn't a specialist in any single mission profile. When configured with the appropriate hardware and software, a small number of UAV platforms might be able to serve a wide variety of different missions effectively, and DDS middleware is key to this multi-mission framework.

Configurable UAV Platforms Reduce Complexity

The success of a configurable UAV platform in fulfilling multiple missions has the potential to affect the entire UAV product life cycle. Design and implementation are likely to be less complex, as the platform is conceptually and technically simpler. Design parameters will change, because of the need to support a larger range of missions, but should become less complex overall.

The biggest advantages are likely to be seen in the operational phase of the life cycle. First, there will be fewer UAV models to support - perhaps two or three platforms rather than a dozen or more different models. This makes provisioning simpler and less expensive, and inventory management becomes less complex. But ultimately it reduces support complexity and costs, because one ground station can serve the platform, rather than requiring separate ground configurations for each separate UAV model. Since the UAV platform by itself is not equipped for any specific mission profile, it is likely that more components and technical support will be required to reconfigure UAV platforms into specific roles and missions. This, however, will be more than offset by the advantages gained in supporting and provisioning fewer UAV models and designing and implementing architecturally simpler UAV platforms.

Integration Framework Connectivity Solves Challenge

Getting to the point where a UAV design can serve as a multi-mission platform is a significant technical challenge. A large part of that challenge is providing an integration framework that supports that concept. Today, UAV implementation is driven by dedicated, hardwired connections between instruments, data collection devices, control surfaces, and the ground control station. While this configuration can guarantee data availability across the components that require data for real-time response, it becomes complex when there are more than a handful of interconnections.

Further, a hardwired configuration is highly resistant to change. For example, if bandwidth requirements were to increase in the future because of new instruments, subsystems, or sensors, it might not be possible to obtain the needed performance or availability guarantees on that UAV. In fact, that is one of the reasons why UAVs tend to be mission-specific.

Conversely, a true platform accommodates full interconnectivity between all possible data sources and consumers. At first glance, that can be enormously complex because there are no preconceived notions of data paths, network bandwidth, or time determinism. Even if such a configuration could be designed and built, there seemingly could be no guarantee that the available bandwidth would meet the actual performance and real-time requirements of any given combination of instruments.

A platform also requires the ability to deliver the necessary bandwidth, performance, and guarantees to any possible configuration. Of course, there are always cases that demand trade-offs in order to achieve this, but the technical trade-offs between performance, reliability, and real-time determinism, for instance, should be both possible and reasonable for the multi-mission integration framework.

Data-Centric Architecture Via DDS

Such an integration framework can be implemented through the use of a software layer, or middleware, that abstracts the details of point-to-point data transmission away from the UAV's individual components. In doing so, this middleware can offer the ability to easily add new hardware and applications, make data available using multiple available routes to ensure real-time availability, support multiple transport protocols, and provide for tuning to specific configurations and missions.

This type of software layer has been created as the Data Distribution Service for Real-Time Systems, a standard published by the Object Management Group. The Data Distribution Service uses the publish-subscribe communications model to enable data producers to publish data to the infrastructure and allow data consumers to subscribe to data from this data infrastructure. The DDS publish-subscribe model automatically connects information producers (publishers) with information consumers (subscribers). The communications are decoupled in space (nodes can be anywhere), time (delivery may be immediately after publication or later), and flow (delivery may be made reliably, tuned to available bandwidth).

In a DDS implementation, the data is abstracted away from the physical source and destination and made accessible to any application that subscribes to it, independent of the source's location and the specific link technology that transports the data (Figure 1). Since DDS coupling is loose and anonymous, communication paths can be defined, created, and destroyed at runtime. Further, DDS implementations automatically "discover" requesting publishers and subscribers, establishing connection between them at runtime with no previous configuration. DDS also enables a resiliency of data necessary for fault tolerance across the network.

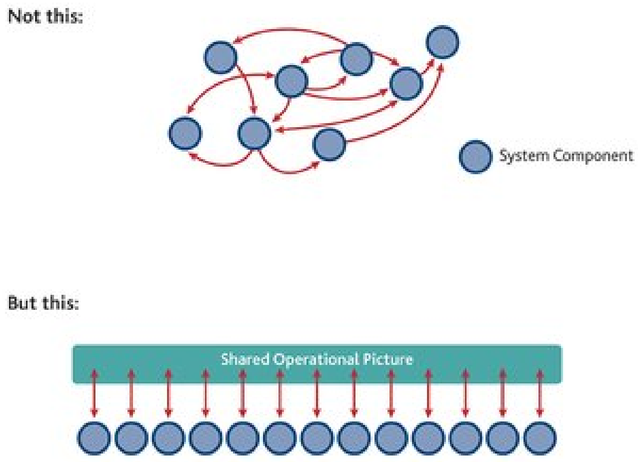

Figure 1

Additionally, in considering our multi-mission UAV platform, the DDS standard defines a comprehensive set of Quality of Service (QoS) parameters. Because there might be design trade-offs, engineers can configure a balance between performance, reliability, determinism, and other factors affecting the system's ability to perform its mission. DDS QoS parameters specify the degree of coupling between components and properties of the overall model and of the data transfers themselves.

DDS includes QoS parameters such as reliability, durability, deadline, priority, and data ownership. By adjusting QoS parameters, system and application software developers will be able to ensure that data transmission and reception meet the unique needs of each system and application. It is possible to change QoS parameter settings at runtime, supporting reconfiguration for a specific mission without the need to rebuild application software. The rich QoS parameter set makes it possible to implement DDS on a wide range of processors and networks, including those that are embedded.

DDS is fundamentally designed to work over unreliable transports, such as UDP or wireless networks. No facilities require central servers or special nodes. All communication, including discovery, is strictly peer-to-peer and optionally employs multicasting for efficiency and scalability.

Example: A Data-Centric Integration Foundation

Consider a UAV that has been designed with a DDS implementation serving as a software integration platform supporting largely interchangeable hardware and applications. Implementing DDS middleware as the foundation for data communication affords a great deal of flexibility on the actual payload. It can focus on sensors and data recording instruments, ensuring that continuous data streams are available to record and analyze data.

Or it can focus on real-time communication, ensuring that telemetry data to and from the ground station is exchanged reliably and deterministically. Once the platform has been created, engineers can look closely at QoS trade-offs to ensure that system data availability requirements are met. QoS settings can be changed at runtime to meet specific mission requirements.

Emerging Uses in UAV Design

DDS is currently used in implementations of both UAVs and ground stations. For example, DDS middleware underlies the General Atomics Advanced Cockpit Ground Control Station (GCS). The GCS networking system integrates controls and information displays, synthetic video, and fused situational awareness data.

The DDS middleware forms the software communications backbone of this station. The DDS publish-subscribe architecture eases system integration for communications. For instance, any system component can subscribe to the incoming aircraft telemetry stream, such as latitude and longitude, pitch, roll, and airspeed parameters. It also allows users to connect multiple workstations, for instance, allowing a pilot at one station to work closely with a sensor operator at a different station.

As another example, the Insitu ScanEagle long endurance UAV uses DDS on the vehicle itself and in the ground control stations (Figure 2). On the airframe, DDS connects the flight computers, sensors, and onboard application computers. Within the ground station, DDS connects the systems that decode data feeds, analyze the UAV's situation, and interface to the operator control. DDS implements a hierarchical control network with well-controlled data flows. This allows Insitu, for instance, to seamlessly switch control between multiple ground stations and to reliably connect to the aircraft over unreliable low-bandwidth links.

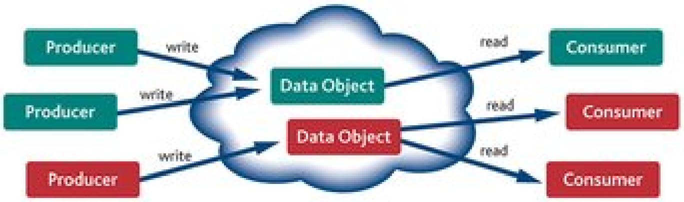

Figure 2

The DDS QoS configurability also makes it well suited to lossy networks such as might be encountered with the wireless connections between the vehicle and ground station.

Of course, DDS doesn't address all of the challenges surrounding the implementation of a flexible UAV platform. Factors such as equipment and payload weight, aerodynamic balance, cost, and power consumption will drive different UAV platforms, even with a flexible and responsive integration model. But given the capabilities of the DDS standard for abstracting and managing real-time data flow, it can offer a fundamental approach to building a multi-purpose UAV platform.

Dr. Edwin de Jong *is director of product management and strategy, core products at Real-Time Innovations, Inc. He has more than 15 years of experience in the architecture and design of large-scale distributed real-time systems. These systems encompass C4I, radar, track management, multi-sensor data fusion, threat evaluation, weapon and sensor assignment, and simulation and training. He coauthored many published technical papers and reports. His fields of expertise include systems architecture, large-scale distributed real-time systems, fault tolerance, networking and communication protocols, real-time in-memory databases, system modeling and analysis, formal specification and analysis, and software engineering systems and tools. Edwin holds a Ph.D. in Mathematics and Physics from Leiden University, The Netherlands. He can be contacted at Edwin.Dejong@rti.com.*