WHITEPAPER

Data Model Considerations for Radar Systems

Executive Summary

The market demands that today's radar systems be designed to keep up with a rapidly changing threat environment, adapt to new technologies, and reflect the realities of smaller development budgets. In this white paper, learn why radar systems developers are adopting standards-based communications technologies and data-centric architectures that help meet these challenging requirements.

The issues discussed relate to the two different approaches that are commonly used for constructing scalable, open radar systems:

- **Integrating radar functions into a single standards-based radar system.** A plug-and-play platform enables the use of devices and subsystems from multiple vendors. It also reduces redundancy by sharing common processing tasks across systems.
- **Networking simple sensors and systems together.** The linked platforms share data in real time using open standards. This approach also enables operational capabilities that standalone systems lack.

The standards-based open architectures for radar and combat systems offer greatly improved flexibility, reduced risk, and lower lifecycle costs.

Architecture Commonality

While the physical designs differ greatly, radar systems incorporate a common, modular, software architecture (see Figure 1).

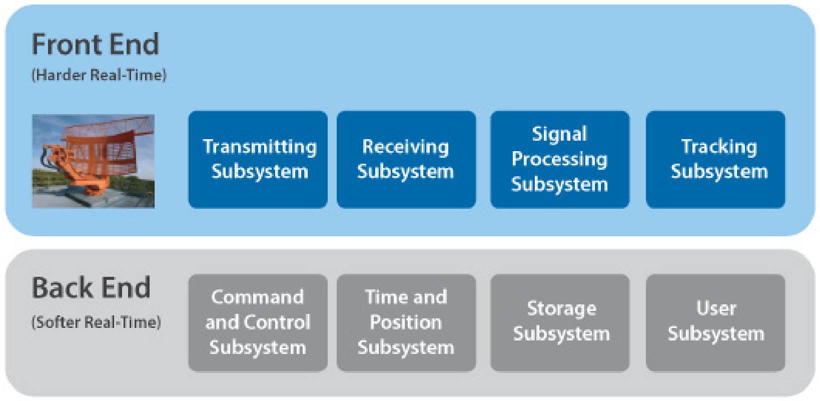

Figure 1: Common software architecture.

The basic subsystems include the antenna (for transmitting radiation and receiving returns), signal processing, and the user interface. Complexity increases along multiple axes depending on additional system physical requirements or deployment use cases. For example, some radar systems are mobile, some are autonomous, others cooperate with external systems, and still others use combinations of these capabilities. As systems increase in scale and complexity, additional architectural components to support these capabilities include command and control (to address variations in system use), time and position (for mobile radar), storage (for ongoing analysis), and detection tracking. All of these subsystems operate in parallel.

The subsystems closest to the transmitter comprise the front end of the radar system, and generally have more stringent real-time requirements. The back-end system components have less stringent real-time requirements. (Note that the time and position subsystem must support all subsystems).

Data Flows

Each subsystem consists of multiple processors and many software components, presenting system architects with internal data flow issues to resolve. In the more complex systems (for example, those with command and control subsystems), traditional point-to-point lines of communication between subsystems pose a major challenge for system architects seeking to adapt their systems to new or emerging requirements. The challenge is compounded with the introduction of any external cooperating system, such as a track fusion or data analysis system.

To greatly simplify data flow compared to traditional point-to-point designs, data-centric communications architectures have been adopted successfully for more recent radar designs. Multiple connections between every subsystem are replaced with a data bus (see Figure 2).

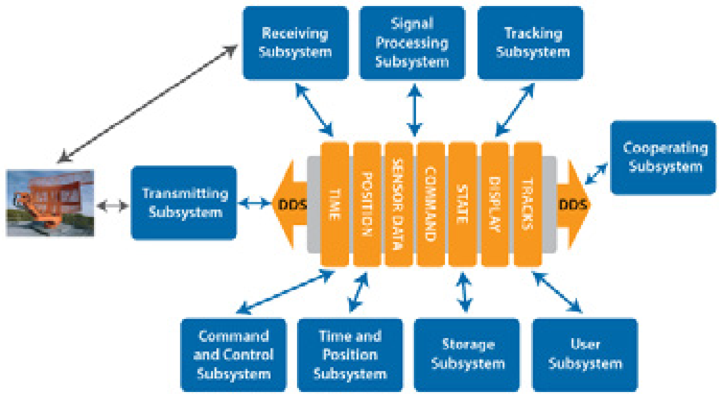

Figure 2. Data-centric communications architecture.

Data-centric architectures simplify radar designs and make them more adaptable. Instead of having software components talking to fixed endpoints in the system (for example, addressing sensor data to another specific physical software component like a tracker), each subsystem or component writes data to a logical data entity incorporated in an information bus, like the open bus enabled by the OMG Data Distribution Service (DDS). This enables location transparency between components, a key enabler for open systems approaches.

Open Systems

With modular and data-centric designs, radar systems lend themselves to open system approaches that in turn help system architects manage the increasing complexities in current designs. The U.S. government is embracing open systems so that multiple organizations can work more cooperatively on the same or separate subsystem components and gain true vendor independence. Some of the key requirements for open systems include:

- Interchangeability (portability). A subsystem from one ship or aircraft must be able to be moved to another ship or plane without complete redesign.
- Ease of upgrades/modification. A new subsystem, with higher capability, should be able to be cost-effectively introduced to an existing system.
- Extensibility. To extend the lifecycle, systems must be able to grow and evolve.
- Ease of integration. Without cumbersome cycles, subsystems should be able to be combined to create new functionality. DDS supports this capability with an open standards-based API and wire protocol that provide vendor interoperability.

Data Distribution Service for Radar Systems

The DDS standards are published and managed by the Object Management Group (OMG), which enables portability across multiple DDS implementations and interoperability between them. The RTI DDS core is based on these standard and open interfaces. The DDS standard, and specifically RTI DDS implementations, span a very wide spectrum of real-time application needs for modular open systems. The technology is unique in its ability to bring low-level information from the system edge to the enterprise and can simultaneously meet the needs for very predictable performance and operation.

Within a DDS environment, interactions between entities are managed in a location-transparent manner using a reliable publish and subscribe software data model. Any new entity, such as an external cooperating system, can join the environment and subscribe to data of interest without requiring the redesign any existing subsystems.

All RTI DDS implementations (Professional, Micro, and Certified) support pluggable transports at the software API level as well as a DDS network wire protocol (see Figure 3).

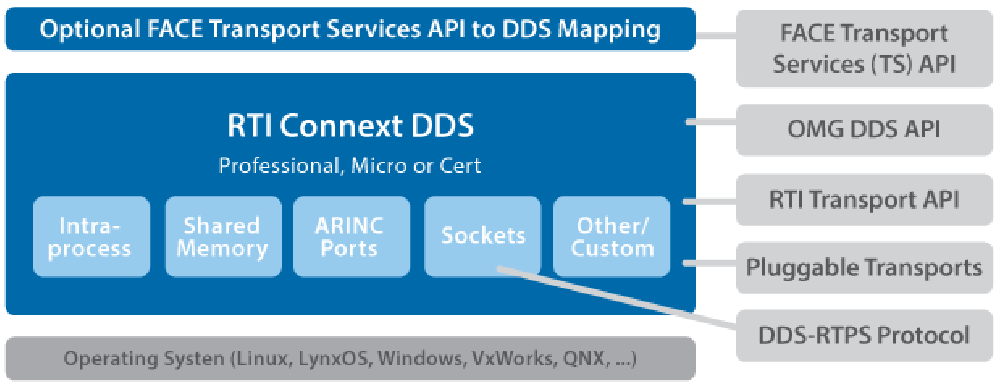
Optional FACE Transport Services API to DDS Mapping
RTI Connext DDS
Professional, Micro or Cert
Intra-process
Shared Memory
ARINC Ports
Sockets
Other/ Custom
Operating Systen (Linux, LynxOS, Windows, VxWorks, QNX, ...)
FACE Transport Services (TS) API
OMG DDS API
RTI Transport API
Pluggable Transports
DDS-RTPS Protocol

Figure 3. Pluggable transports for bridging and mediating between busses, and the optional FACE transport layer for onboard avionics systems.

The RTI DDS implementation provides optimized communications with a rich set of physical transports (see Table 1). Applications can use multiple transports concurrently. As a result, RTI DDS supports the broadest range of proximity and distribution options (see Table 2).

Table 1. RTI DDS Transports.

Transport	Use
Intra-Process	Within the same address space (process)
Shared Memory	Between processes in the same partition
ARINC Ports	Within a node; within or between partitions
Sockets (UDP Unicast or Multicast)	Within or between nodes, including over Ethernet
Low-Bandwidth	Over satellite or radio links (no IP requirement)
Custom	Over custom networks or busses (via plugin API)

Table 2. Connection Mechanism Comparison

		RTI DDS	CORBA	Sockets	POSIX Queues	Shared Memory	Queuing Ports	Sampling Ports
Proximity	Intra-partition	●	●	●	●	●	●	●
	Inter-partition	●	●	●			●	●
	Inter-node	●	●	●				
	Multiple concurrently	●						
Distribution	One-to-one	●	●	●	●	●	●	●
	One-to-many	●		○		●	●	●
	Many-to-one	●		○	●			
	Many-to-many	●		○				

Interoperability and Mediation

Consider the case of providing tracking data to a new external subsystem. The original tracking data might be relative to a ship, or relative to a position on the ground. Making use of the tracking data in the context of an external system means the point of reference for the tracking data must be provided (and this point itself may be in motion). Without the positional information for the entities being tracked, the radar is all but useless.

A DDS data mediation solution can be introduced to accommodate needs of both the providing and consuming systems as shown in Figure 4.

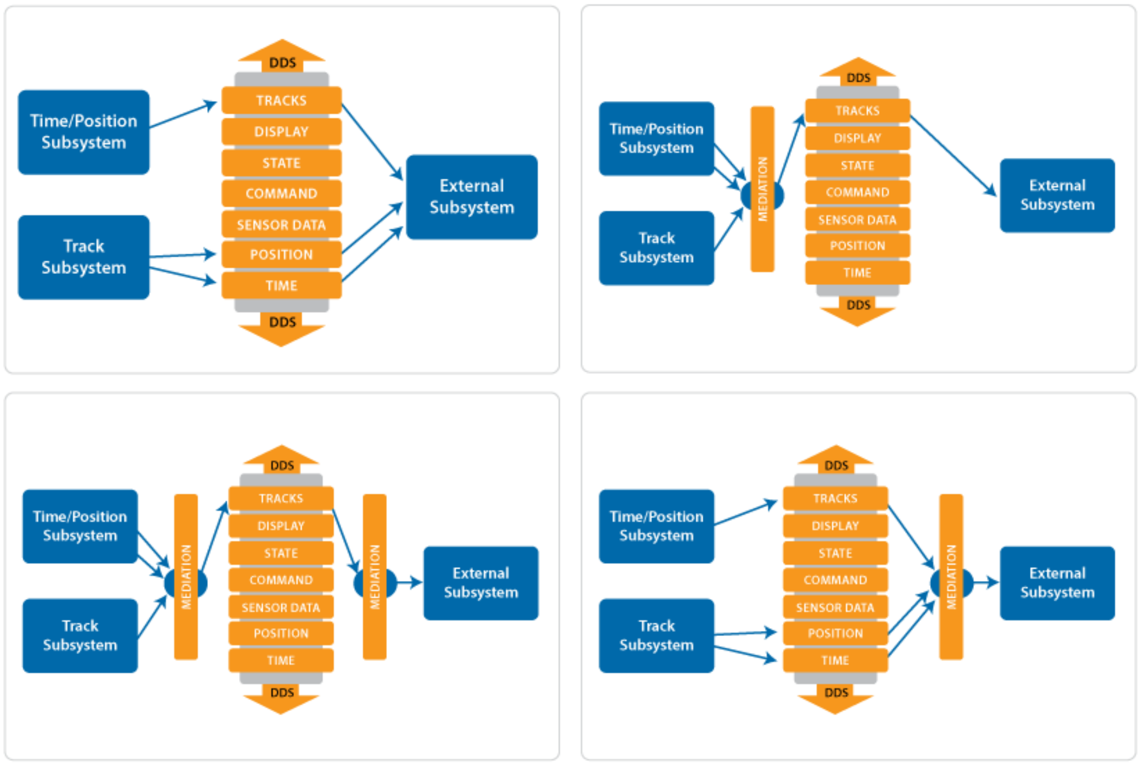

Figure 4. Mediation options for data-centric interoperability: no mediation (top left), publishing mediation (top right), publishing and subscribing mediation (bottom left), and subscribing mediation (lower right).

Mediation can be introduced to combine the time/position data with the tracking data, prior to publishing that information. This can accommodate very high data rates for publishing tracking data. Alternatively, after tracking and positioning data is published separately, mediation can be introduced to take the independently published data and combine it in a manner that is appropriate for the external subsystem. This might be preferable when an external subsystem needs to gather data from many different tracking stations, and needs mediation to keep up with the high publish rates of dozens of radars.

The flexibility to introduce multiple mediation approaches can extend the life of the overall system. Fine-grained data flow control can adjust for changing requirements.

RTI provides a Routing Service for performing mediation. The Service supports multiple transports and data flows between different transports while simultaneously performing any necessary data conversions as they cross the Routing Service. This resulting mediation also enables dynamic data flow creation as new components are discovered in the system, which is increasingly important in complex systems that must support components being continually added and removed.

Data Distribution Design Patterns

A few data design patterns are common in radar systems, each mapping to common categories (archetypes) of radar data. See Figure 5.

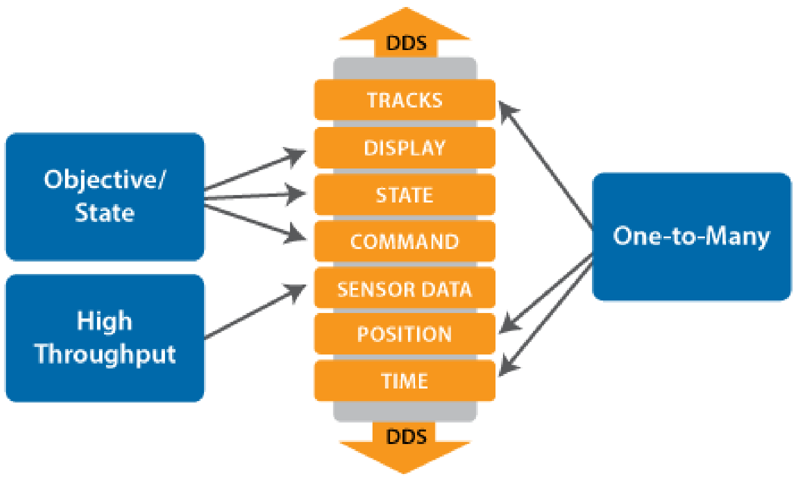

Figure 5. Three common data design patterns for radar systems.

Objective/State Design Pattern

Display, state, and command data are at the heart of the **objective/state** design pattern in radar systems. This pattern deals with the classic radar data design issues. In many systems, there is no differentiation between the commanded (or intended) state and the actual state of an endpoint, but in reality there are three possible states of any endpoint such as a transmitter: the current state, objective (or intended) state, and requested objective state. In systems at rest, the objective state is the same as the requested objective state. In a dynamic system, the two can differ and must be observed accurately.

Within an objective/state system, there are two or more roles. Effectors (for example, Transmitter or Receiver subsystems) provide current state and objective state commands and observe the requested objective state. Requesters (for example, Command and Control subsystems) provide the requested objective state and observe both the current and objective states. There can also be third parties in the system, typically acting as observers of state.

With DDS, an effector will write the current and objective state, and read the requested objective state (see Figure 6). DDS quality of service (QoS) characteristics support durability for the state data, and DDS can also automate historical logging of state data.

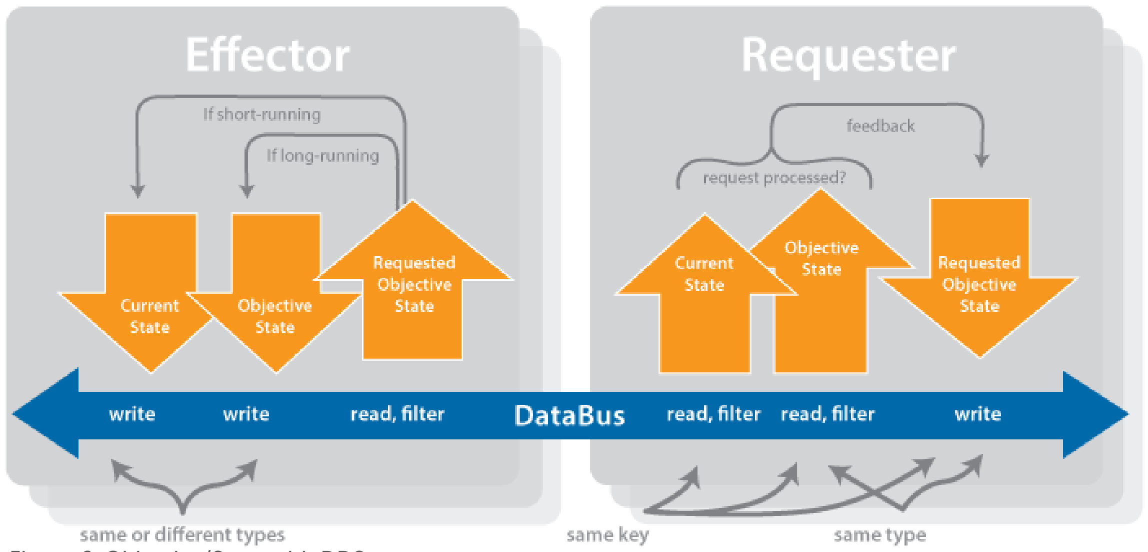

Figure 6. Objective/State with DDS.

One-to-Many Design Pattern

Tracking, position, and time comprise a classic **one-to-many** design pattern in radar systems, which is one of the original DDS data design patterns. The ability to reliably multicast tracking data or sensor data is a great benefit within many types of radar systems. Multicasting offloads the providers of data and is therefore essential for performance. Communicating to many consumers at the same time reduces costs, lowers network traffic, and lowers latency with fewer socket sends.

On the negative side, one-to-many designs can be less reliable or "best-effort" in nature. DDS can improve this aspect of one-to-many designs, with configurable QoS options, but the reliability gains come at a cost (see Figure 7). Compared to unicast, the benefits greatly outweigh the negatives, however.

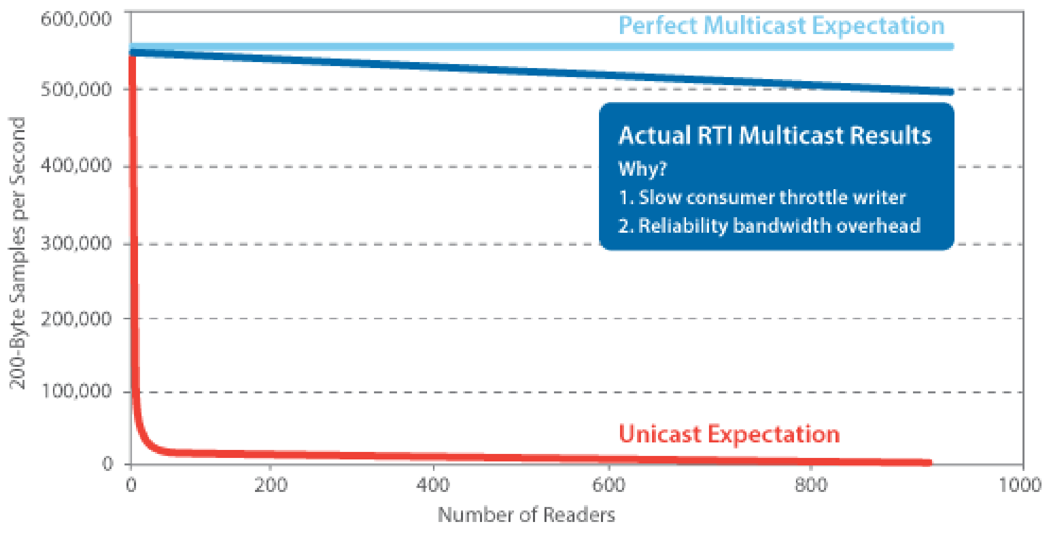

Figure 7. Reliability in a one-to-many data design.

High-Throughput Design Pattern

High-performance sensor equipment requires a **high-throughput** design, with an emphasis on the aggregation of data on the network. The design issues with this type of data pattern include the sampling rates (whether samples arrive continuously or at a high periodic rate) and transport saturation. To accommodate high-throughput periodic data, RTI offers several capabilities:

- Synchronous sends (default)
- Batching
- Multiple reliability paradigms (explicit acknowledgement of all samples, acknowledgment-only protocol for reducing potential load)
- Processing of received data directly in receive thread or application thread, for managing performance on the receiving side

To manage high-throughput data patterns over constrained networks, RTI DDS supports:

- Configurable message sizes
- Batching in a manner that reduces protocol header overhead
- A low-bandwidth network plugin with header and data compression
- A "multi-channel" feature to send data over different network interface cards depending on the data content

Achieving reliability within high-throughput systems requires the management of multiple behaviors, all of which are enabled by RTI DDS solutions:

- Writers can keep data for potential retransmission
- Unpredictable latency can be managed
- Reader behaviors can be monitored and managed
- Data loss can be addressed with a choice of behaviors (declare failure and stop, report error and continue, delay writing for readers to catch up, ignore, etc.)

Choosing the Best Data Model

Data models range from high-level descriptions of data and associations down to the low-level models of the data transport. At the application platform level, the data model provides the context and semantics that are critical for interoperability. The online messages can be compact, depending on how much data is at rest and how much is in motion. Radar systems are highly optimized for the data in motion to be distributed efficiently and the data at rest to be managed separately.

Other data model considerations for radar systems include the extremely long lifecycle of the systems, much longer than consumer product cycles. Radar system designers must actively design to accommodate ongoing change. Multiple refresh cycles must be expected and open architectures and standards are vital for cost containment during the entire lifecycle.

As proven in countless deployments, a data-centric architecture cost-effectively meets all of the above objectives for radar systems. Specifically, DDS inherently supports interoperable radar data models on the wire. The OMG standard provides an Interface Design Language (IDL), which forms the basis for the native DDS data model schema and provides type safety and heterogeneous interoperability across languages, operating systems, and CPUs.

Perhaps most important, DDS allows radar system designers to remain focused on their domain expertise, whether that is mechanical design, algorithms, or custom hardware design. At the heart of constantly changing radar systems, RTI DDS communication middleware isolates system-specific software from processor and network changes. It enables a data-centric architecture that shields developers from the impacts of the changes it very cost-effectively accommodates.

To Learn More

RTI Connext Secure DDS, the world's first turnkey DDS security platform, conforms to the OMG specification and provides a vital security infrastructure that is data-focused for DDS and legacy systems. Data-centric configuration policies make it possible to tailor security to a broad range of content and use cases, and a standards-based optional plugin SDK further enables the alignment of data security with system-specific tools and capabilities.

For more information about RTI Connext and how to leverage it to create more flexible, cost-effective radar systems, visit the RTI web site.

To get started with DDS, download a free trial of the RTI Connext DDS solution.

rti
Your systems. Working as one

CORPORATE HEADQUARTERS
232 E. Java Drive
Sunnyvale, CA 94089

Tel: +1 (408) 990-7400
Fax: +1 (408) 990-7402
info@rti.com

www.rti.com