Addressing the Challenges of Mission-Critical Information Management in Next-Generation Net-Centric Pub/Sub Systems with OpenSplice DDS

Douglas C. Schmidt and Hans van't Hag
PrismTech Corporation,
6 Lincoln Knoll Lane, Suite 100, Burlington, MA, 01803, USA
{doug.schmidt, hans.vanthag@}prismtech.com

Abstract

An important class of new and existing applications requires cooperating entities to share and exchange information seamlessly in real-time, while meeting stringent requirements for quality of service (QoS) and heterogeneity. This paper describes how the recently adopted OMG DDS standard, and the OpenSplice DDS implementation *of this standard, support the data distribution and management challenges posed by next-generation distributed systems, and systems of systems.*

1. Introduction

Recent trends in net-centric systems motivate the development of mission-critical information management capabilities that ensure the right information is delivered to the right place at the right time to satisfy quality of service (QoS) requirements in heterogeneous environments. These information management systems, such as Air Traffic Management Control, Traffic Monitoring and Control, and Management, Railway Control Systems, increasingly run in net-centric environments characterized by thousands of platforms, sensors, decision nodes, and computers connected together to exchange information, support sense-making, enable collaborative decision making, and effect changes in the physical environment.

For example, modern supervisory control and data acquisition (SCADA) systems, air traffic management (ATM) systems, and the Global Information Grid (GIG) [1] are net-centric environments designed to ensure that different services and partners, as well as individuals participating to specific missions, can collaborate effectively and deliver appropriate decision-making and control information to users in a timely, dependable, and secure manner. Achieving this vision requires the distributed middleware to address key design and operational challenges, including ensuring the right data is available at the right place at the right time all the time in heterogeneous environments consisting of dynamically formed coalitions of nodes that can share a common shared state (often referred as the operational picture) and exchange data seamlessly.

To support these mission-critical information management capabilities in net-centric systems, the Object Management Group (OMG) has adopted the Data Distribution Service (DDS) specification [2], which is a standard for QoS-enabled data-centric publish/subscribe (pub/sub) communication that enables net-centric mission- and business-critical information management systems to share information in real-time by utilizing QoS-driven publish/subscribe patterns. Open*Splice* DDS is a DDS-compliant implementation that is used in a wide range of military and commercial systems, including naval combat management systems, commercial air traffic control, transportation management, automated stock trading systems, and semiconductor fabrication devices.

This paper describes why Open*Splice* DDS is an important distributed software technology for mission-critical net-centric systems because it supports (1) *relational* and *object oriented modeling* of the information shared by, and distributed within, the system (2) *location independence*, via anonymous publish/subscribe protocols that enable communication between collocated or remote publishers and subscribers, (3) *scalability*, by supporting large numbers of topics, data readers, and data writers and platform portability and (4) *interoperability*, via standard interfaces and transport protocols.

The remainder of this paper is organized as follows: Section 2 presents an overview of the OMG DDS discussing the DDS QoS policies that are the most relevant for net-centric mission-critical information management systems; Section 3 provides an overview of OpenSplice DDS, explain its key architectural features, and provide a performance characterization; Section 4 explains how Open*Splice* DDS has been applied in practice to address key challenges of developing and operating distributed software in current and planned net-centric mission-critical information management systems; and Section 5 concludes the paper by describing current trends and future areas of extension.

2. DDS Overview

DDS provides the following capabilities for net-centric mission-critical information systems:

- **Universal access to information** from a wide variety of sources running over potentially heterogeneous hardware/software platforms and networks,
- **An orchestrated information bus** that aggregates, filters, and prioritizes the delivery of this information to work effectively under the restrictions of transient and enduring resource constraints,
- **Continuous adaptation to changes** in the operating environment, such as dynamic network topologies, publisher and subscriber membership changes, and intermittent connectivity, and
- **Standard QoS policies and mechanisms** that enable applications and administrators to customize the way information is delivered, received, and processed in the appropriate form and level of detail to users at multiple levels in net-centric systems.

This section describes the key capabilities and entities in DDS and shows how its QoS policies can be used to specify and enforce performance-related requirements of net-centric mission-critical information management systems. Figure 1 shows the various profiles and layers in the DDS standard. The lower layer defines a Data-Centric Publish Subscribe (DCPS) platform, whose goal is to provide efficient, scalable, predictable, and resource aware data distribution. The higher layer is the Data Local Reconstruction Layer (DLRL), which is an optional interface that provides an object-oriented facade atop the DCPS. The DLRL extends the DCPS to allow object-oriented (OO) information models to specify the information shared by a DDS application. .

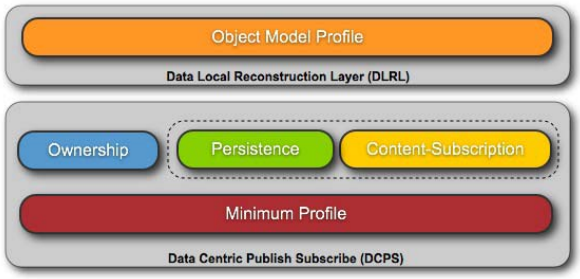

Figure 1: Profiles and Layers in the DDS Standard

A separate specification, called the Real-Time Publish/Subscribe (RTPS) DDS interoperability wire protocol [3], defines the standard network protocol used to exchange data between publishers and subscribers that use different implementations of DDS. The remainder of this section describes the conceptual model of DDS and explains the QoS policies that are most relevant for net-centric mission-critical information management systems.

2.1 DDS Conceptual Model

Domains and partitions. DDS applications send and receive data within a domain. Domains can be divided into partitions that allow the separation and protection of different data flows. Although DDS entities can belong to different domains, only participants within the same domain can communicate, which helps isolate and optimize communication within communities that share common interests. For example, each communication layer within a large-scale distributed system could be associated with a DDS domain, and further subdivided into partitions. This approach isolates domain participants across layers, which enables effective use of resources and helps enforce security and confidentiality policies.

Global data space. DDS provides a strongly typed global data space within each domain in which applications produce and consume dynamically changing portions of a shared information model, as shown in Figure 2. DDS's information model capabilities are similar to those of relational databases, except that DDS's global data space is completely distributed, QoS-aware, and allows anonymous and asynchronous sharing of a common information model. The DDS information model is the only knowledge publishers and subscribers need to communicate, *i.e.*, they need not be aware of each other nor be concerned with low-level network programming details, such as IP addresses, port numbers, remote object references, or service names. By allowing data to flow where and when needed, DDS's global data space enables the sharing of mission-critical information and situational awareness needed to implement net-centric mission-critical information management systems.

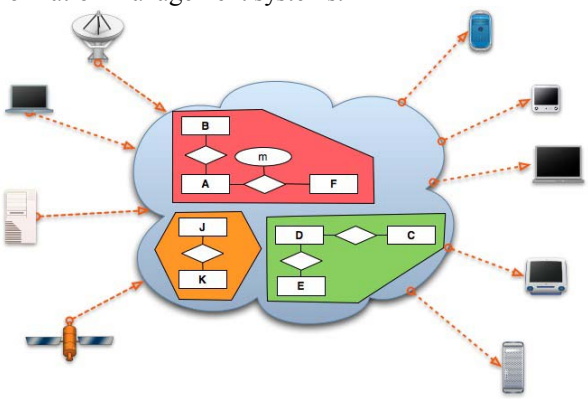

Figure 2: DDS Global Data Space in a Domain

Topic. A DDS topic is an association between a data type, a set of QoS policies, and a unique name, as shown in Figure 3. A topic is also the unit of information contained in DDS's global data space and is used by applications to define their information model and associate QoS policies with it. DDS applications in

net-centric systems define their information model by identifying topics that are relevant for solving their requirements and organizing them into either relational or object-oriented models. DDS allows the expression of the system information model as either (1) a topic relational model, which can be thought as an extension of the familiar entity relationship diagrams used in data bases, decorated with QoS, or (2) an object-oriented model, which can also be synthesized as an object-oriented view of the relational model.

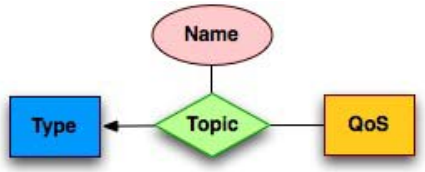

Figure 3: DDS Topic

The DCPS layer provides support for relational modeling, while the DLRL extends the DCPS with an object-oriented facade, so that applications can either completely ignore the DCPS relational models or build an object models atop the DCPS. When using the DLRL layer, familiar OO constructs, such as inheritance and relationships, can be used to model the information to share and distribute within the system, further improving expressiveness, productivity, and reusability. Data associated with DDS topics are expressed using types defined by the standard OMG Interface Definition Language (IDL), which simplifies the inter-working between DDS and CORBA. Relationships between topics can be captured via keys that can be used to distinguish between different instances of the same topic. In net-centric mission-critical information systems, an information model will be associated with every layer in which DDS-based data exchange occurs. This information model, which can comply with open standards, is the *lingua franca* used by the different applications in coalitions to exchange information and seamlessly interoperate. Likewise, the QoS policies decorating the information model determine how the data is disseminated, persisted, and received in the global data space.

Publishers and subscribers. In net-centric mission-critical information management systems, publishers and subscribers correspond to a range of domain participants, such as embedded devices, air traffic control radars, visualization consoles, and online stock feeds, as well as planning and simulation services in operations centers. DDS applications use data writers to publish data values to the global data space of a domain and data readers to receive data. A publisher is a factory that creates and manages a group of data writers with similar behavior or QoS policies, as shown in Figure 4. A subscriber is a factory that creates and manages data readers, as also shown in Figure 4.

Publishers can declare their intent to produce data on particular topics with associated QoS, and provide the data on those topics to the global data space. Subscribers receive topic data from the global data space that match their subscriptions (the rules that define what represents a matching subscription are described below). Configuring QoS policies allows publishers and subscribers to (1) define the local behavior of their interface with DDS, such as the number of historical data samples they require and the maximum update-rate at which they want to receive data, and (2) how DDS should provide global data availability with respect to reliability, urgency, importance, and durability of the distributed data. Since topics themselves can be annotated with QoS policies that drive the global data-availaibity, publishers and subscribers can use these pre-defined QoS policies as consistent defaults allowing them to concentrate purely on the local-behavior driving policies.

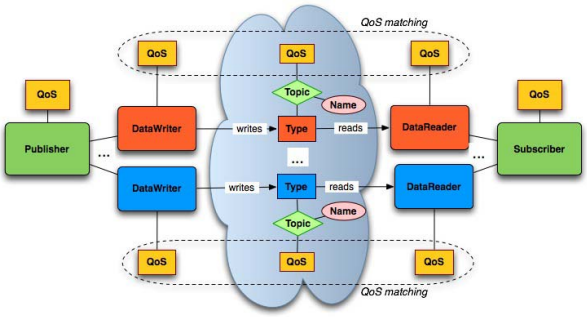

Figure 4: DDS Publisher/Writer Subscriber/Reader and Subscription/QoS Matching

Subscriptions and matching. A subscription is an operation that associates a subscriber to its matching publishers, as shown in the center of Figure 4. In addition to the topic-based subscriptions described above, DDS also supports *content-based subscription*, in which a subset of the standard Structured Query Language (SQL) is used to specify subscription filters. In DDS a *matching* subscription must match the following two types of topic's properties: (1) its features, such as its type, name, key, and content, and (2) its QoS policies, which are described in the *QoS Policies* section.

The matching process for QoS uses a requested/offered (RxO) model shown in Figure 5, where the requested QoS must be less than or equal to the offered QoS. For example, subscribers requesting reliable data delivery cannot communicate with publishers that only distribute data using best effort delivery. Likewise, subscribers cannot request a topic update

whose deadline is smaller than that declared by any publishers.

The subscription matching mechanism provided by DDS enforces a powerful form of "design by contract" [5], where QoS is used together with type information to decide whether publishers and subscribers can communicate. This extended form of design by contract helps ensure that net-centric systems will operate as intended, both from functional and QoS perspectives. These assurances are essential in the development, deployment, and operation of mission-critical net-centric mission-critical information management systems.

Discovery. Another key feature of DDS is that all information needed to establish communication can be discovered automatically, in a completely distributed manner. Applications dynamically declare their intent to become publishers and/or subscribers of one or more topics to the DDS middleware, which uses this information to establish the proper communication paths between discovered entities. This capability supports dynamic scenarios common in net-centric mission-critical information management where cooperating domain participants join and leave throughout system operation.

2.2 Quality of Service (QoS) Policies

DDS is designed for mission-critical net-centric systems where the right answer delivered too late becomes the wrong answer. To meet this requirement it is essential that the middleware controls and optimizes the use of resources, such as network bandwidth, memory, and CPU time. Figure 5 shows the rich set of QoS policies that DDS [2] provides to control and constrain topic (T), data reader (DR), data writer (DW), publisher (P), and subscriber (S) resources and QoS properties, such as persistence, reliability, timeliness, and memory usage. Below we discuss the DDS QoS policies that are the most relevant for net-centric mission-critical information management systems.

Data availability. DDS provides the following QoS policies that control the availability of data to domain participants:

- The DURABILITY QoS policy controls the lifetime of the data written to the global data space in a domain. Supported durability levels include (1) volatile, which specifies that once data is published it is not maintained by DDS for delivery to late joining applications, (2) transient local, which specifies that publishers store data locally so that late joining subscribers get the last published item if a publisher is still alive, (3) transient, which ensures that the global data space maintains the information outside the local scope of any publishers for use by late joining subscribers, and (4) persistent, which ensures that the global data space stores the information persistently so it is available to late joiners even after the shutdown and restart of the system. Durability is achieved by relying on a durability service whose properties are configured by means of the DURABILITY_SERVICE QoS of non-volatile topics.
- The LIFESPAN QoS policy controls the interval of time during which a data sample is valid. The default value is infinite, with alternative values being the time-span for which the data can be considered valid.
- The HISTORY QoS policy controls the number of data samples (i.e., subsequent writes of the same topic) that must be stored for readers or writers. Possible values are the last sample, the last n samples, or all samples.

These QoS policies provide the DDS global data space with the ability to cooperate in highly dynamic environments characterized by continuous joining and leaving of publisher/subscribers. This capability makes it possible for net-centric mission-critical information management systems to share a common operational picture regardless of the dynamism that characterizes portions of the systems, such as coalitions of soldiers collaborating in urban environments or coordinated UAVs in support of mission-critical operations.

QoS Policy	Applicability	RxO	Modifiable	
DURABILITY	T, DR, DW	Y	N	Data Availability
DURABILITY SERVICE	T, DW	N	N	
LIFESPAN	T, DW	-	Y	
HISTORY	T, DR, DW	N	N	
PRESENTATION	P, S	Y	N	Data Delivery
RELIABILITY	T, DR, DW	Y	N	
PARTITION	P, S	N	Y	
DESTINATION ORDER	T, DR, DW	Y	N	
OWNERSHIP	T, DR, DW	Y	N	
OWNERSHIP STRENGTH	DW	-	Y	
DEADLINE	T, DR, DW	Y	Y	Data Timeliness
LATENCY BUDGET	T, DR, DW	Y	Y	
TRANSPORT PRIORITY	T, DW	-	Y	
TIME BASED FILTER	DR	-	Y	Resources
RESOURCE LIMITS	T, DR, DW	N	N	
USER_DATA	DP, DR, DW	N	Y	Configuration
TOPIC_DATA	T	N	Y	
GROUP_DATA	P, S	N	Y	

Figure 5: Key QoS Policies for Net-centric Systems

Data delivery. DDS provides the following QoS policies that control how data is delivered and how publishers can claim exclusive rights on data updates:

- The PRESENTATION QoS policy gives control on how changes to the information model are presented

to subscribers. This QoS gives control on the ordering as well as the coherency of data updates. The scope at which it is applied is defined by the access scope, which can be one of INSTANCE, TOPIC, or GROUP level.

- The RELIABILITY QoS policy controls the level of reliability associated with data diffusion. Possible choices are RELIABLE and BEST_EFFORT distribution.
- The PARTITION QoS policy gives control over the association between DDS partitions (represented by a string name) and a specific instance of a publisher/subscriber. This association provides DDS implementations with an abstraction that—if implemented properly—segregates traffic generated by different partitions, thereby improving overall application scalability and performance.
- The DESTINATION_ORDER QoS policy controls the order of changes made by publishers to some instance of a given topic. DDS allows the ordering of different changes according to source or destination timestamps.
- The OWNERSHIP QoS policy controls which writer 'owns' the write-access to a topic when there are multiple writers and ownership is EXCLUSIVE. Only the writer with the highest OWNERSHIP_STRENGTH can publish the data. If the OWNERSHIP QoS policy value is shared, multiple writers can concurrently update a topic. OWNERSHIP thus helps to manage replicated publishers of the same data.

These QoS policies control the reliability and availability of the data, thus allowing the delivery of the right data to the right place at the right time. More elaborate ways of selecting the right data are offered by the DDS content-awareness profile that allows applications to select information of interest based upon their content.

Data timeliness. DDS provides the following QoS policies that control the timeliness properties of distributed data:

- The DEADLINE QoS policy allows applications to define the maximum inter-arrival time for data. DDS can be configured to automatically notify applications when deadlines are missed.
- The LATENCY_BUDGET QoS policy provides a means for applications to inform DDS of the urgency associated with transmitted data. The latency budget specifies the time period within which DDS must distribute the information. This time period starts from the moment the data is written by a publisher until it is available in the subscriber's data-cache ready for use by reader(s).
- The TRANSPORT_PRIORITY QoS policy allows applications to control the importance associated with a topic or with a topic instance, thus allowing a DDS implementation to prioritize more important data relative to less important data.

These QoS policies make it possible to ensure that mission-critical information needed to reconstruct the shared operational picture is delivered in a timely manner.

Resources. DDS defines the following QoS policies to control the network and computing resources that are essential to meet data dissemination requirements:

- The TIME_BASED_FILTER QoS policy allows applications to specify the minimum inter-arrival time between data samples, thereby expressing their capability to consume information at a maximum rate. Samples that are produced at a faster pace are not delivered. This policy helps a DDS implementation optimize network bandwidth, memory, and processing power for subscribers that are connected over limited bandwidth networks or which have limited computing capabilities.
- The RESOURCE_LIMITS QoS policy allows applications to control the maximum available storage to hold topic instances and related number of historical samples

DDS's QoS policies support the various elements and operating scenarios that constitute net-centric mission-critical **information** management. By controlling these QoS policies it is possible to scale DDS from low-end embedded systems connected with narrow and noisy radio links, to high-end servers connected to high-speed fiber-optic networks.

Configuration. The QoS policies described above, which provide control over the most important aspects of **data** delivery, availability, timeliness, and resource usage. In addition, DDS also supports the definition and distribution of user specified bootstrapping information via the following QoS policies:

- The USER DATA QoS policy allows applications to associate a sequence of octets to domain participant data readers and data writers. This data is then distributed by means of the DCPSParticipant built-in topic. This QoS policy is commonly used to distribute security credentials.
- The TOPIC_DATA QoS policy allows applications to associate a sequence of octet with a topic. This bootstrapping information is distributed by means of the DCPSTopic built-in topic. A common use of this QoS policy is to extend topics with additional information, or meta-information, such as IDL type-codes or XML schemas.

- The GROUP_DATA QoS policy allows applications to associate a sequence of octets with publishers and subscribers. This bootstrapping information is distributed by means of the DCPSSubscription,and DCPSPublication built-in topics, respectively. A typical use of this information is to allow additional application control over subscriptions matching.

3. OpenSplice DDS

Open*Splice* DDS is an implementation of the OMG DDS v1.2 characterized by high performance, high availability and extreme scalability and predictability. This section explains how these properties are provided by Open*Splice* DDS's architectural features that have been designed to support scalability, predictability, and optimal resource usage. Figure 6 shows Open*Splice* DDS' macroscopic organization, showing how it can run on both enterprise operating systems, such as Windows, Linux, and Solaris, as well as embedded real-time operating systems, such as VxWorks and LynxOS. Open*Splice* DDS relies on a set of modular services that support networking, resource management, persistence, database and Web Services integration. On top of these services its DCPS and DLRL implementations, are exposed in several different language bindings, such as C, C++, and Java. Additional high-level services include monitoring, configuring and logging systems running on Open*Splice* DDS.

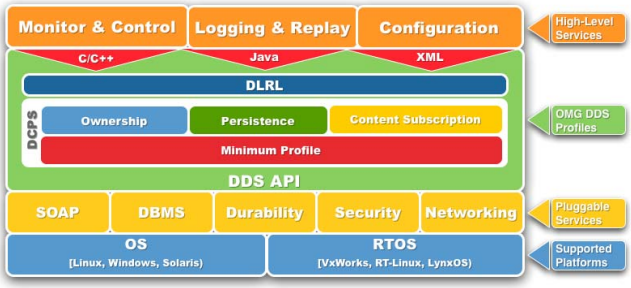

Figure 6: Open*Splice* DDS Features

Architectural Highlights. The Open*Splice* DDS architecture, shown in Figure 7, is built around a few core concepts: (1) efficient usage of shared resources, such as memory and networking to warrant proper scalability, (2) QoS driven management of these resources to provide maximal determinism, and (3) a pluggable-service architecture that both implements essential DDS functionality, such as priority-pre-emptive networking or distributed fault-tolerant durability, as well as allows for flexible adaptation of the middleware to balance functionality and footprint.

Scalability and Efficiency. To reduce code-footprint, a single shared-library provides the OpenSplice DDS binding (API's) to both applications and its pluggable services. This library provides efficient sharing of information within a single host by utilizing a ring-fenced shared memory segment holding a single-copy of each shared data-item regardless of the number of applications that have expressed interest in that data. For efficient dissemination of information between hosts, the OpenSplice DDS Networking service schedules access to the shared network based upon the actual urgency (the LATENCY_BUDGET QoS) of the information.

Determinism and Safety. Critical resources, such as memory and network interfaces, are managed by OpenSplice DDS to provide the determinism as required in realtime systems and bound the impact of misbehaving applications in mission-critical systems. For example, DDS QoS policies for information importance (TRANSPORT_PRIORITY) and scope (PARTITION) drive the priority pre-emptive data distribution by the networking service. This trusted service also shields potentially misbehaving applications from the network.

Pluggable services. Within a single host, the Open*Splice* DDS shared memory segment provides and mediates application access to the various pluggable Open*Splice* DDS services, such as the networking service, the durability service, and DBMS integration service. Applications will thus read and write data to the shared memory, while the services will access the same shared memory segment concurrently to perform their activities.

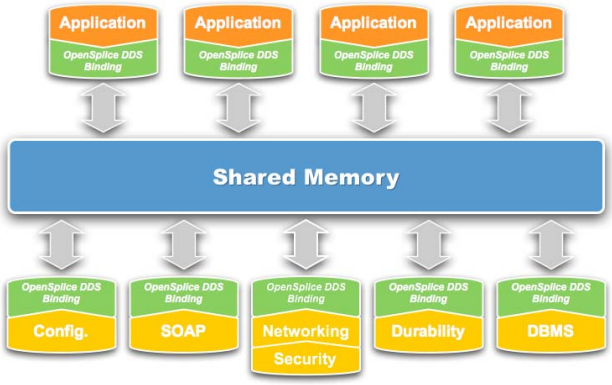

Figure 7: Open*Splice* DDS Architecture

Networking Service. The Open*Splice* DDS networking service represents one of its most distinguishing features. The networking service reads data from the shared memory and ships it to subscribers in other computing nodes making optimal use of resources whilst enforcing the required QoS levels. Thanks to the shared memory architecture, (1) the overhead of local communication is minimized due to the presence of a shared memory that already interconnects all applications, (2) applications do not have direct access to the

networking resources, so the network service can enforce transport priorities across applications, make inter-application data bundling, and perform more effective traffic shaping, and (3) when multiple applications on a node subscribe to the same topic this architecture can minimize memory usage and marshaling overhead.

The Open*Splice* networking service supports any user-defined number of network channels dedicated to handle traffic of a specified priority range. This feature enforces messages priority even on non-priority-preserving transports, such as the TCP/IP or UDP/IP. Another important feature provided by the Open*Splice* DDS network service is traffic shaping. For every channel it is possible to define the traffic profile, and ensure that the network utilization never exceeds a user-specified value.

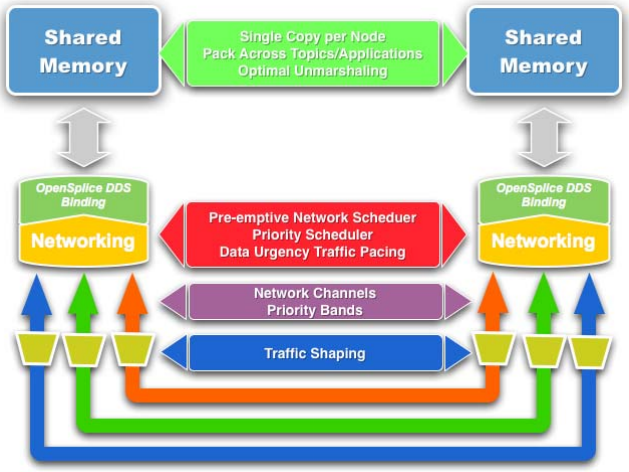

Figure 8: Open*Splice* DDS Networking Service

4. DDS Success Stories

Standard adoption. Although DDS is a relatively new standard (adopted by the OMG in 2004), it has been adopted quickly. Its fast paced adoption stems from its ability to address key requirements of data distribution in net-centric systems, as well as the maturity and quality of available implementations, which are based on decades of experience developing data-centric middleware for mission-critical systems.

DDS has been mandated by (1) EuroControl as the standard publish/subscribe mechanism for exchanging flight data plans between Air Traffic Control Centers, (2) the US Navy's Open Architecture Computing Environment [6] as the standard publish/subscribe technology to use in next-generation combat management systems, and (3) DISA as the standard technology for publish/subscribe to be used in all new or upgraded systems [7]. Several major mission-critical programs, such as the European Next Generation Flight Data Processor, CoFlight, US Navy's DDG-1000 land attack destroyer, US Army's Future Combat Systems (FCS), and the Thales TACTICOS Combat Management System also adopted DDS even before it was mandated, underscoring DDS's ability to address the data distribution challenges of next-generation mission-critical information management systems.

OpenSplice DDS use cases. Open*Splice* DDS was one of the most influential technologies contributing to the definition of the DDS standard, due in large part to its heritage of success during the past two decades in a range of mission-critical distributed systems. In particular, Open*Splice* DDS has been deployed in several different application domains, such as defense, aerospace, transportation, etc. For example, the TACTICOS combat management system [8] developed by THALES Naval Netherlands builds atop Open*Splice* DDS [9], which allows TACTICOS to achieve very high scalability, from small ships to aircraft carrier grade, as well as high performance, availability, and determinism even under temporary overload conditions. TACTICOS is currently in use in over15 Navies worldwide serving more than 26 ships-classes ranging from small patrol boats up to large frigates.

The utilization of Open*Splice* DDS is instrumental in the success of TACTICOS since it provides both the scalability to support thousands applications running on hundreds of distributed computers on a typical frigate size system. Another key feature of Open*Splice* DDS is its 'damage resistance,' i.e.,if an error occurs on a specific computer, DDS can dynamically re-allocate application software to the remaining computer pool. The DDS Persistence Profile as supported by Open*Splice* DDS is instrumental in this dynamic reallocation since it allows applications to store their internal state into the DDS middleware, which manages this state in a distributed and fault-tolerant way so that reallocated applications can regain their state after restart and can continue what they were doing before they crashed.

Open*Splice* DDS supports a data-centric approach where at the start of the system-design, the information model can be captured, annotated with proper QoS policies, and then shared between multiple parties. This federated architecture is common in existing and planned 'coalition-based' developments where multiple parties jointly implement the overall combat system. Open*Splice* DDS provides the fault tolerant information backbone onto which all these applications are deployed and is thus responsible for providing each application with the right information at the right time.

Along with the rapid adoption of Open*Splice* DDS in the defense domain, its use is also steadily growing in other domains, such as transportation, telecommunications, and finance. For example, in the context of Air Traffic Control (ATC) and Management (ATM),

Open*Splice* DDS has been selected as the publish/subscribe middleware for distributing flight data plans in CoFlight [12], which is the next generation European Flight Data Processor. In general, Open*Splice* DDS is an appropriate middleware technology for application domains that require rich support for QoS policies and high-performance and dependability standards-based, commercial-off-the-shelf (COTS) implementations.

5. Concluding Remarks

Open*Splice* DDS is standards-based QoS-enabled data-centric pub/sub middleware that provides a feature rich data-centric real-time platform to support the needs of current and planned net-centric mission-critical information management systems. Its powerful set of QoS policies—together with its scalable architecture [4]—makes it an effective and mature choice for solving the data distribution and information management problems net-centric systems. Below we summarize how Open*Splice* DDS addresses the key challenges outlined in Section 1 in a standard and interoperable manner:

- **Shared operational picture**. Open*Splice* DDS provides effective support for these types of applications via its QoS policies for defining the scope, content, and QoS of the data model that underlies the operational picture.
- **The right data at the right time at the right place** via Open*Splice* DDS QoS policies that enable a fine-grained control over information delivery, such as the ability to control many aspects of data dissemination to ensure timely delivery and optimal resource usage.
- **Heterogeneous environment**. By providing standard QoS policies that control the bandwidth used for providing data to interested parties, Open*Splice* DDS runs in heterogeneous platforms while providing different elements with a common operational picture.
- **Dynamic coalitions**. The highly dynamic nature of Open*Splice* DDS, such as its support for dynamic discovery, provides an effective platform for supporting ad hoc interactions.

From a standardization perspective, DDS continues to evolve to meet new operational and technical challenges of net-centric mission-critical information management systems. The OMG is currently pursuing the following three types of extensions for DDS:

- The first involves adding new platform-specific models that fully leverage programming language features, such as standard C++ containers.
- The second extension deals with extensible topics that enable incremental system updates by ensuring that changes in the data model do not break interoperability.
- The third set of extensions focus on network data representation and the syntax used to define topics. For example, upcoming versions of the DDS standard will likely allow the definition of topics using XML, as well as the use of XML or JSON as the network data representation. DDS security has not yet been standardized.

References

1 The United Stated Department of Defense Quadrennial Defense Review Report, February 2006, www.defenselink.mil/qdr/report/Report20060203.pdf

2 OMG, "Data Distribution Service for Real-Time Systems Specification," www.omg.org/docs/formal/04-12-02.pdf

3 OMG, Real-Time Publish Subscribe Protocol – DDS Interoperability Wire Protocol Specification www.omg.org/cgi-bin/apps/doc?ptc/06-08-02.pdf.

4 Ming Xiong, Jeff Parsons, James Edmondson, Hieu Nguyen, and Douglas C. Schmidt, "Evaluating Technologies for Mission-critical Information Management in Net-Centric Systems," Proceedings of the Defense Transformation and Net-Centric Systems conference, April 9-13, 2007, Orlando, Florida.

5 Bertrand Meyer, Object Oriented Software Construction - 2nd Edition, Prentice Hall, 2001.

6 Open Architecture Computing Environment, www.nswc.navy.mil/wwwDL/B/OACE.

7 Defense Systems Information Agency, DoD Information Technology Standards Registry, https://disronline.disa.mil.

8 THALES, "TACTICOS Combat Management System, Exploiting the Full DDS Potential," www.omg.org/docs/dds/06-12-06.pdf.

9 OpenSplice DDS, www.prismtech.com/openplice-dds.

10 OMG DDS SIG Portal, portals.omg.org/dds.

11 OMG DDS Forum, www.dds-forum.org.

12 CoFlight eFDP, www.omg.org/docs/dds/07-07-04.pdf.