Event Notification Service for Interconnected ATM Systems

Christian Esposito
Dipartimento di Informatica e Sistemistica - Universita' di Napoli Federico II
Via Claudio 21, 80125 - Napoli, Italy
christian.esposito@unina.it

Abstract

In the context of mission critical systems, we are witnessing to a paradigm shift from a proprietary and embedded solutions to open and modular systems of systems. In this scenario, data dissemination software infrastructures are playing a crucial role, since they have to guarantee interoperability among different and heterogeneous systems by satisfying at same time real-time and fault-tolerance requirements. Furthermore, the communication infrastructure has to meet stringent security requirements, due to its critical role played into the deployment of the next generation of mission critical systems. Publish/subscribe communication paradigm has recently received considerable attention to cope with these issues, and OMG DDS is a standard (carried out by the Object Management Group) that follows this interaction paradigm. The simultaneous satisfaction of multiple QoS requirements exposes the OMG DDS as an encouraging solution to the flight information distribution. However, the standard does not address security issues as well as data heterogeneity ones. The authors Ph.D. research focuses on the evaluation of OMG DDS, with the purpose to assess the benefits of its use as flight information distribution system. To this aim, OMG DDS issues will be analyzed and evaluated in order to investigate novel data distribution algorithms and software infrastructures to let Data Distribution Services ready to be integrated in the next generation of mission critical systems.

1 Problem Statement

In the past few years, mission critical systems have undergone a serious architectural evolution, moving from monolithic systems, to highly modular architectures. The challenge is to federate a constellation of systems, making a fully integrated system of systems. As a practical example of this direction, we can consider the roadmap outlined by EuroControl for the European Air Traffic Management (ATM) evolution. In order to handle more efficiently the growing aviation traffic [1], the solution is the interconnection of all the European ATM systems by a middleware framework. The pivotal issue to be addressed is the realizing of an information dissemination infrastructure, which is able to exchange different typed data among different kind of interconnected systems.

The middleware must be proven to satisfy a set of simultaneous Quality-of-Service (QoS):

- **Timeliness**. Typically, mission critical operations show a time-critical behavior: failing in delivering information out of time boundaries could lead to instability, which might result in threats to either infrastructures or human lives.
- **Fault-tolerance**. Mission critical systems must be dependable and able to handle error conditions properly. The information dissemination infrastructure must expose high measure of availability and reliability, showing the ability of continue operation when a fault occurs.
- **Scalability**. The rising of the activity of a single system and the escalation of connected systems could increase the number of data exchanges, and consequently the time of computation. To avoid this problem, the middleware has to be gracefully designed for scalable data dissemination.
- **Data Heterogeneity**. The existence of a common data presentation (e.g. XML) does not imply common semantics. Users and developers can properly interpret and use collected data only when they have adequate context information. In traditional systems, this context information is typically left implicit and is generally lost when data crosses a component or institutional boundaries.
- **Security**. Whilst it is essential that legitimate participants have rapid and efficient access to the essential information, it is necessary to safeguard that information from misuse by terrorists, subversive organizations and

Abstraction	Space Decoupling	Time Decoupling	Synchronization Decoupling
Message Passing	No	No	Producer-side
RPC/RMI	No	No	Producer-side
Asynchronous RPC/RMI	No	No	Yes
Future RPC/RMI	No	No	Yes
Notification	No	No	Yes
Distributed Shared Memory	Yes	Yes	Producer-side
Message Queuing	Yes	Yes	Producer-side
Publish/Subscribe	Yes	Yes	Yes

Decoupling Abilities of Interaction Paradigms

Figure 1: Comparison of communication paradigms

similar. Security in this context refers not only to the preventing the unauthorized reading of data, but further to preventing unauthorized or malicious writing, modification or deletion of information.

Moreover, the QoS degree has also to be guaranteed when the systems are joined on a geographical scale by wide-area networks. The simultaneous satisfaction of these requirements involves conflict and compromise, and it is still a trial for modern middleware implementations [2].

2 Background and Open Issues

Among the wide scope of proposed middleware architectures, publish/subscribe paradigm [3] is an emerging interaction scheme, that provides decoupling both in time, space and synchronism among the two participants of a communication, as shown in fig. 1. Decoupling is a desirable property, because it furthers scalability. Publish/subscribe allows an asynchronous many-to-many communication pattern; this is a very interesting characteristic for two reasons. First, many-to-many connections allow a scalable dissemination of informations and let the middleware become manageable and flexible. Then, an asynchronous middleware will not block the application for processing: the request can regularly continue processing, regardless of the state of the other applications or occurred faults. The mission critical systems make extensive use of distributed computing over wide-area and heterogeneous networks, and publish/subscribe is a promising solution in achieving such a distributed computing.

A significant amount of publish/subscribe systems have been developed and implemented, both by industry and by academia [4]. It is possible to group these different implementations according to:

1. **adopted architecture**, which affects scalability. It can be *centralized*, i.e. a server placed between publishers and subscribers; *distributed*, i.e. communication primitives implementing a store and forward mechanisms both on the producer and on the consumer sides; and *federated*, i.e. a distributed network of servers.

	Centralized	Distributed	Federated
Topic-based	GREEN / JEDI	HERALD / JMS	JEDI
Content-based	GREEN / CORBA NS	DREAM	REBECA / SIENA
Type-based	GREEN	OMG DDS	HERMES

Figure 2: Comparison of Publish/Subscribe implementations

2. **subscription model**, which affects end-to-end delivery time. It can be *topic-based*, i.e. participants can publish and subscribe to individual subjects; *content-based*, i.e. events are classified according to a matching function on their content; and *type-based*, i.e. events belongs to a specific type, encapsulating attributes as well as methods.

In fig. 2, it is shown a taxonomy of some existing publish/subscribe middlewares, according to the two outlined dimensions. Most of these middlewares have lacked the support necessary for mission critical systems [5]. The main weaknesses of these solutions are related to either a limited or not existing support for QoS, either to the lack of architectural properties which promote dependability and survivability. Recently, in order to fill this gap, Object Management Group developed a new standard, called Data Distribution Service (DDS) [6]. The purpose of the specification is to provide a common application-level interface for interoperable publish/subscribe middleware and targets real-time communications, balancing predictable behavior and implementation efficiency/performance. It relies on the use of different QoS to tailor the service to the application requirements [7]. In particular QoS parameters allow to tune the robustness of the middleware against the network unavailability and the information timeliness. To maintain the desirable decoupling as much as possible, the detail for QoS Policy follows the subscriber-requested / publisher-offered pattern. In this pattern, the subscriber side can indicate a ”requested” value for a particular QoS Policy. The Publisher side names an ”offered” value for this QoS Policy. If the two positions are compatible, then communication will be established. A first benchmarking work [8] demonstrated that OMG DDS-compliant implementations perform significantly better than other publish/subscribe implementations. This is due in part to the fact that OMG DDS decouples the information intent from information exchange.

The current OMG DDS is implemented on a simple overlay structure, where a publisher is directly linked with all the interested subscribers. This solution is simple and efficient, however, when the system has to scale to wide-area networks, this basic overlay organization presents obvious scalability limits. Intrinsically the publish/subscribe interaction scheme does not still address two key open issues. First, most implementations, e.g. OMG DDS, do not support interactions among heterogeneous participants [8]. Second, security is not addressed at all [9]. Aside from granting access to the system only at authorized participants, it is difficult to decide whether the whole infrastructure is trustworthy or none, due to the anonymous nature of the publish/subscribe. In accord to these considerations, four properties has to be assured, in order to make OMG DDS a fit technology for the drafted integration issue:

- **WAN Scalability**: ability to scale in the setting of a wide-area network;
- **Confidentiality**: absence of unauthorized disclosure of information;
- **Integrity**: lack of improper information alterations;
- **Semantic Metadata Sharing**: support to an explicit description of the intended meaning of message content.

3 Approach

The author's Ph.D. research project focuses on the assessing the benefits of using OMG DDS as the solution for the data distribution middleware, and in particular to ATM integration. The challenge is to realize a benchmarking evaluation of performance and dependability properties of OMG DDS over wide-area infrastructures. Currently, several vendors provide their own OMG DDS implementation. Each implementation is characterized by additional services and proprietary extensions to the standard. In [10], it is demonstrated that individual OMG DDS implementations are optimized for different use cases. The author aims to develop test sessions of the three different OMG DDS implementation, e.g. RTI DDS, OpenSliceDDS and TAO DDS. These tests have to analyze behavior of these architectures in several testbed and show their eventual weaknesses. A Field Failure Data Analysis (FFDA) campaigns and error injection experiments will infer a fault model of OMG DDS. The understanding and evaluation of OMG DDS behavior will be conducted also using an operational model to predict its behavior in a complex situation, to avoid time-consuming analysis. The effect of this integrated evaluation step is the degree of timeliness and fault-tolerance achievable by OMG DDS, and its satisfying of distribution infrastructure requirements.

The author will propose three extensions, in order to overcome the limitations of the OMG DDS standard. First, a solution to WAN scalability problem is to architect the system as a distributed network of server. Epidemic algorithms have recently gained popularity as an effective solution for information dissemination in wide-area systems [11]. In addiction to their inherent scalability, they are easy to deploy, robust and resilient to failure. The author means to estimate the respect of the data distribution requirements by the exiting algorithms of group communication. In the event that these algorithms are found unsuitable, a novel data dissemination algorithm has to be considered. Second, an integrity check and data encrypting mechanisms will be implemented, in order to earn the right security degree. Third, additional meta-data will be introduced, in oder to identify the semantic nature of the content of exchanged messages. All the proposed techniques have to be characterized by a low overhead in terms of delivery time, so to satisfy the timeliness.

References

[1] EuroControl. Eatms operational concept document, ver. 1.1.

[2] P. Narasimhan. Trade-offs between real-time and fault-tolerance for middleware applications. *Workshop on Foundations of Middleware Technologies*, November 20002.

[3] R. Guerraoui P.Th. Eugster, P. A. Felber and A. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2):114 – 131, January 2003.

[4] S. Tarkoma and K. Raatikainen. State of art review of distributed event systems. *MinemaEventsReport C0-04*, February 2006.

[5] L. Querzoni R. Baldoni, A. Milani and S. Tucci-Piergiovanni. Exploring dds specification limits in network-centric scenarios. *Technical Report - MIDLAB Universit di Roma*, February 2007.

[6] Object Management Group. Data distribution service (dds) for real-time systems, v1.2. *OMG Document*, 2007.

[7] G. Pardo-Castellote. Omg data-distribution service: Architectural overview. *Proceedingsof the 23rd International Conference on Distributed Computing Systems Workshops*, pages 200 – 206, May 2002.

[8] C. Bornhvd M. Cilia, M. Antollini and A. Buchmann. Dealing with heterogeneous data in pub/sub systems: The concept-based approach. *Proceedings of International Workshop on Distributed Event-based Systems (DEBS 2004)*, pages 26 – 31, 2004.

[9] D. Evans C. Wang, A. Carzaniga and A. Wolf. Security issue and requirements for internet-scale publish-subscribe systems. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, pages 303 – 311, 2002.

[10] R.E.Schantz and D.C. Schmidt. Research advances in middleware for distributed systems. *Proceedings of the IFIP 17th World Computer Congress - TC6 Stream on Communication Systems: The State of the Art*, 220:1 – 36, 2002.

[11] A. Kermarrec P. T. Eugster, R. Guerraoui and L. Massouliea-cute. Epidemic information dissemination in distributed systems. *COMPUTER - IEEE Computer Society*, 37(5):60 – 67, May 2004.