A Meta-Model Transformation between DDS and DBMS Representation of Data for DDS-DB Integration

Hyung-Jun Yim[1], Yun-Hee Son[2], Chan-Ho Song[1], Chumsu Kim[2], Gyoonjung Lee[2] and Kyu-Chul Lee[1†]

[1]*Department of Computer Engineering, Chungnam National University*
220 Gung-Dong, Yuseong-Gu, Daejeon, Korea
[2]*Naval Combat Systems PEO, Agency for Defense Development,*
Jinhae-Gu, Changwon-Si, Gyeongnam, Korea
[1]*{hyungjun25, mellow211, chanho, kclee}@cnu.ac.kr* , [2]*{chskim, gjlee}@add.re.kr*

Abstract

Data Distribution Service for Real-Time Systems (DDS) is the OMG publish/subscribe standard that aims to enable scalable, real-time, dependable, high performance and interoperable data exchanges. DDS introduces a virtual Global Data Space where applications can share information by simply reading and writing data-objects addressed by means of an application-defined name (Topic) and a key. Large numbers of publishers and subscribers on DDS can create huge amounts of Topic data. The commonly accepted approach to large-scale real-time data management is to use a database for persistence.

This paper provides an integral solution for data distribution and database management in the real-time applications space. This paper outlines a mapping between DDS and DBMS meta-model and DDS-DB integration mechanism for supporting data persistence. The prototypes of mapping tools have been implemented. This paper also concludes with general remarks and a discussion of future works.

Keywords: *Data Distribution Service, DBMS, Integration, Meta-Model, IDL, mapping*

1. Introduction

Data Distribution Service (DDS) [1] is a Publish/Subscribe technology for real-time, dependable and high-performance data exchanges. Publish/Subscribe is an abstraction for one-to-many communication that provides anonymous, decoupled, and asynchronous communication between a publisher and its subscribers.

The Object Management Group (OMG) DDS standards family is today composed by the DDS v1.2 API and the DDS Interoperability Wire Protocol (DDSI v2.1), supporting the Data Centric Publish/Subscribe (DCPS) layer, and the Data Local Reconstruction Layer (DLRL). DDS applications can communicate even if they are written in different programming languages or running on different operating systems or processor architectures. And because DDS defines standard programming interfaces, application software is also portable across different DDS implementations. The DDS API standard defines several different profiles that enhance real-time Publish/Subscribe with content filtering, persistence, automatic fail-over, and transparent integration into object-oriented languages.

Real-time data in DDS must be captured, stored, retrieved, queried, and managed such that

† Corresponding Author

the proper information can be quickly accessed by all interested participants within the system. This data management capability can be viewed as, but is not limited to, a distributed real-time database where peer-to-peer (P2P) networking and in-memory data management systems are leveraged to provide a solution that manages storage, retrieval, and distribution of fast changing data in dynamic network environments [2].

One of the main characteristics of DDS, in addition to ensuring deterministic data exchange, is providing mechanisms for the management of many Quality of Service (QoS) aspects to meet application requirements. DDS features fine and extensive control of QoS parameters, including reliability, bandwidth, delivery deadlines, and resource limits. In contrast with message-oriented paradigms, instead of exchanging data in the form of messages, DDS uses a platform independent way to define the data to be sent inherited from the CORBA specification known as Interface Definition Language (IDL) [3]. This approach allows the automatic generation of communication source code for specific target platforms [4].

In this paper, we provide a mapping to bridges the impedance mismatch between DDS and DBMS representation of data. The DDS-DB integration can bridge data from the DDS domain to the DBMS domain and vice versa. In DDS, data is represented by topics, while in DBMS data is represented by tables. With DDS-DB integration, a mapping between a topic and a table can be defined. A topic in DDS will be mapped to a table in DBMS. When data in the DDS domain has to be available in the DBMS domain, the DDS-DB integration can be configured to facilitate that functionality.

To support these functions, we propose a real-time data communication and management mechanism for database. DDS implements some of the best practices followed by specific real-time data distribution middleware products developed by companies like RTI DDS [5] and Thales. Although initially there were few DDS products, nowadays the number of DDS vendors is increasing and there even exist some open source products, such as PrismTech OpenSplice [6], OpenDDS [7], BEE DDS [8], ETRI DDS (EDDS), Twin Oaks CoreDX DDS [9] and Real-Time Communication Middleware (ReTiCom) [10]. We refer our proposed framework to DB-ReTiCom based on ReTiCom with FastDB [11]. DB-ReTiCom is built as a highly modular collection of pluggable service that provide database integration with any ODBC-compliant DBMS.

This paper is organized as follows: the next section introduces the related works and a mapping between DDS and DBMS meta-model is given in Section 3. In Section 4, we propose a description of DDS-DB integration mechanisms. Lastly, in Section 5, the conclusions and future works related to this paper.

2. Related Works

RTI and OpenSplice support a bridge to integrate representation of global data space in DDS and DBMS. RTI Real-Time Connext [12], as seen in Figure 1, is the integration of two complementary technologies: DDS and DBMS. This powerful integration allows your applications to uniformly access data from real-time/embedded and enterprise data sources via the DDS API (such as with RTI DDS) or via the SQL API. Since both these technologies are data-centric and complementary, they can be combined to enable a new class of applications. In particular, DDS can be used to produce a truly decentralized distributed DBMS, while DBMS technology can be used to provide persistence for DDS data.

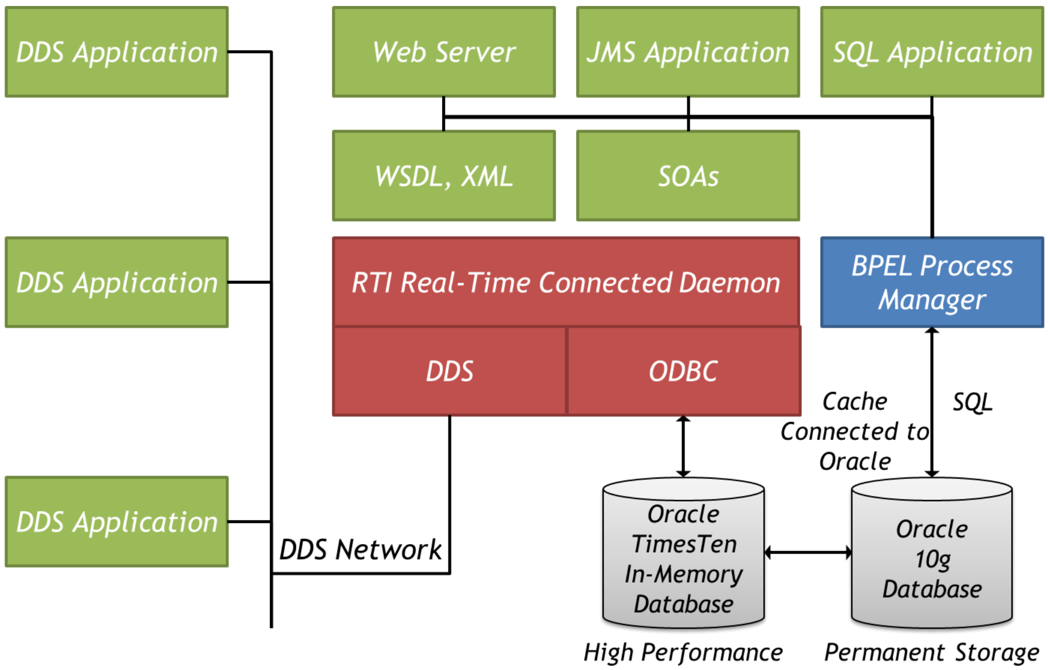

Figure 1. RTI Real-Time Connect Bridges Embedded and Enterprise Applications

The OpenSplice DbmsConnect [13] Module is a pluggable service of OpenSplice that provides a seamless integration of the real-time DDS and the non-/near-real-time enterprise DBMS domains, as shown in Figure 2. It complements the advanced distributed information storage features of the OpenSplice Persistence Module (and vice versa). Where (relational) databases play an essential role to maintain and deliver typically non- or near-real-time 'enterprise' information in mission systems, OpenSplice targets the real-time edge of the spectrum of distributing and providing 'the right information at the right place at the right time' by providing a QoS aware 'real-time information backbone'.

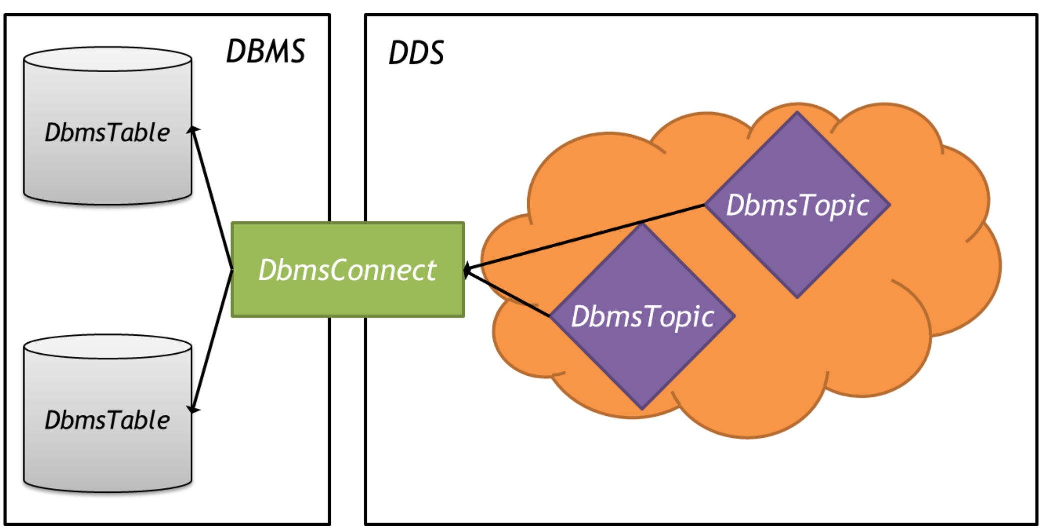

Figure 2. DDS to DBMS Scenario

3. Mapping between DDS and DBMS Meta-Model

This section describes the meta-model and data type mapping that DDS-DB integration uses to connect DB-ReTiCom to FastDB databases. In DDS data is called a "Topic" and represents the unit of information that can be produced or consumed by a DDS application. A Topic is defined as a triad composed of by a type, a unique name, and a set of QoS policies which are used to control the non-functional properties associated with the Topic. Topic Types can be represented with the subset of the OMG IDL standard that defines struct types, with the limitations that Any-types are not supported. The only thing that DDS has in common with CORBA is that it uses a subset of IDL. FastDB is a highly efficient object relational in-memory database system with real-time capabilities and convenient C++ interface.

3.1. Meta-Model Transformation between DDS and DBMS

The DDS concept of a keyed topic and a type is mapped to the DBMS notion of a keyed table and schema representing data-object instances. Rules are specified for translating between a DBMS table record and the DDS wire format representation as listed in Table 1.

Table 1. DDS-DBMS Meta-Model

	DB-ReTiCom	FastDB
API	Accessed via DDS API (C/C++ language bindings)	Accessed via SQL-like syntax API (C/C++ language bindings)
Data Structure	IDL	Table Schema
Identification	Topic (identified by a name string)	Table (identified by a name string)
Data Values	Typed Data	Rows in Table
Key	Instance Key	Special Field (using as Primary Key)
Write	DDSDataWriter::write()	SQL INSERT or UPDATE
Dispose	DDSDataWriter::dispose()	SQL DELETE

The complex data types available in both domains map onto each other as listed in Table 2. A Topic type is a struct that can contain as fields nested structures, unions, enumerations, template types as well as primitive types. The sequence type with respect to its length and contained type.

Table 2. Complex Data Type Mapping

IDL	XSD	C++	FastDB
interface	mapping to get, set	class	class
Struct	complex type	struct	struct
Enum	simple type	enum	enum
sequence	complex type	class	dbArray<t>
array	complex type	array	dbArray<t>
inheritance	○	○	○

Data in FastDB is stored in tables which correspond to C++ classes whereas the table records correspond to class instances. C++ types described in [11] are accepted as atomic components of FastDB records. The basic data types available in both domains map onto each other as listed in Table 3. The string and wstring can be parametrized only with respect to their maximum length.

Table 3. Basic Data Type Mapping

IDL	XSD	C++	FastDB
short (2byte)	short (2byte)	short (2byte)	Int2 (2byte)
unsigned short (2byte)	unsigned short (2byte)	unsigned short (2byte)	Nat2 (2byte)
long (4byte)	int (4byte)	int (4byte)	Int4 (4byte)
unsigned long (4byte)	unsigned int (4byte)	unsigned int (4byte)	Nat4 (4byte)
long long (8byte)	long (8byte)	long long (8byte)	Int8 (8byte)
unsigned long long (8byte)	unsigned long (8byte)	unsigned long (8byte)	Nat8 (8byte)
float (4byte)	float (4byte)	float (4byte)	Real4 (4byte)
double (8byte)	double (8byte)	double (8byte)	Real8 (8byte)
long double (10byte)	double (8byte)	long double (10byte)	Real8 (8byte)
char (1byte)	string	char (1byte)	char const*
wchar (2byte)	string	wchar_t (2byte)	wchar const*
string	string	string	std::string
wstring	string	wstring	std::wstring
boolean (1byte)	boolean (1byte)	boolean (1byte)	char (1byte)
octet (1byte)	unsigned byte (1byte)	unsigned char (1byte)	Byte (1byte)

3.2. Mapping Tools

This paper describes the 3 mapping tools for meta-model transformation between DDS and DBMS as seen in Figure 3 and Figure 4.

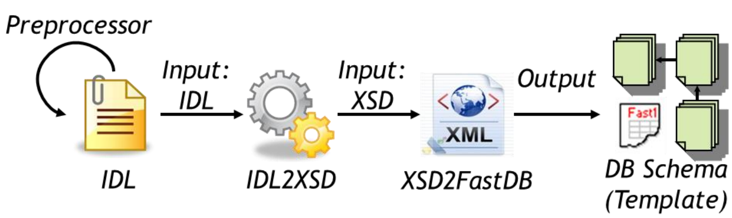

Figure 3. The Steps of Meta-Model Mapping Tools

Our preprocessor is a program that processes its input data (user-defined IDL) to produce output that is used as input to IDL2XSD tool. IDL2XSD is an IDL compiler to W3C XML Schema. IDL2XSD generates the XML schema for the supplied IDL. XSD2FastDB is an XSD compiler to FastDB schema. XSD2FastDB generates an identified by a name string (Topic.cpp) template, to demonstrate how generated source can consumed.

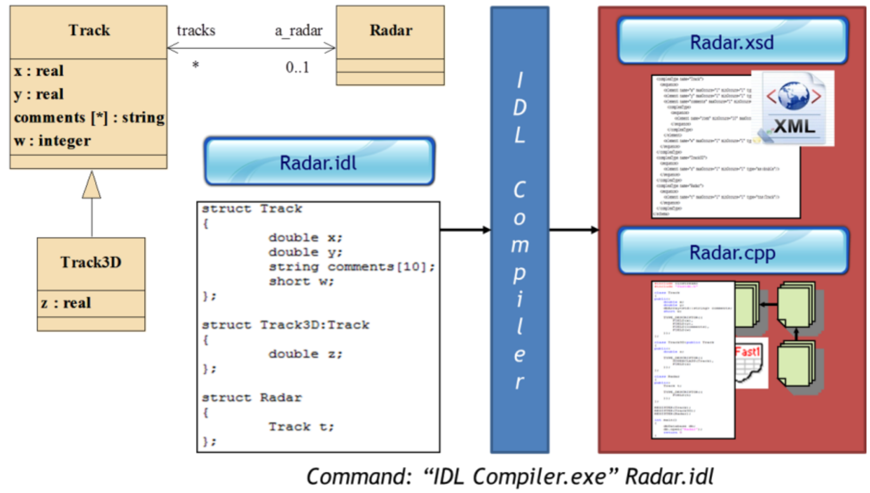

Figure 4. Radar Topic Example of Mapping Tools

4. DDS-DB Integration Mechanism

We propose a DB-ReTiCom based on the DDS and persistent data space with functional capability of dynamic reconfiguration. Data space is built on an in-memory database system in order to provide high-performance and real-time support. The DDS-DB Integration mechanism not only manages the automatic publication of changes made to tables in the DBMS but also apply changes received via DDS to tables in the DBMS. DDS Topic in database, making it available to late-joining subscribers even if the original publisher is no longer accessible. And web and enterprise applications can interface to real-time applications through a database or using web services interfaces, including WSDL/SOAP and REST.

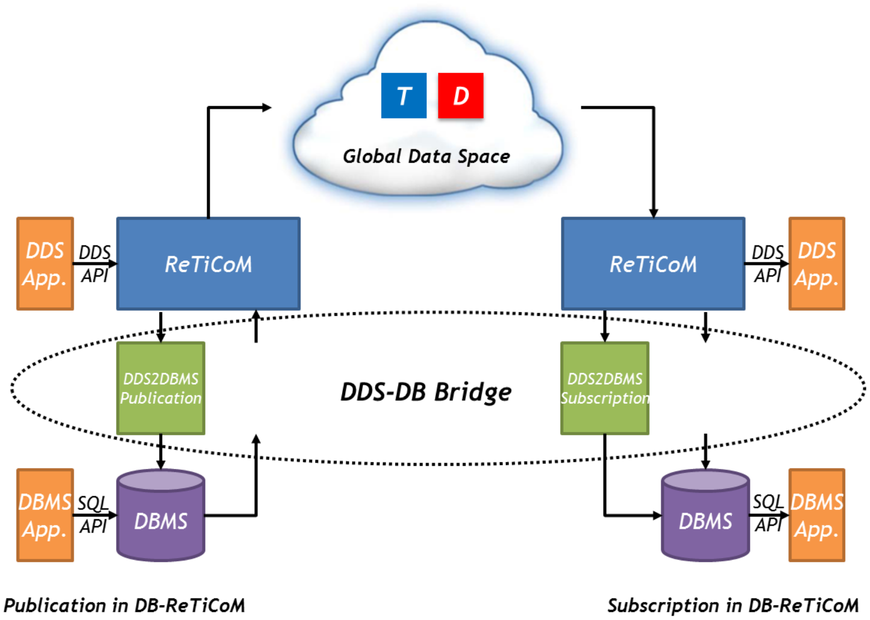

Figure 5. DDS-DB Integration Mechanism

4.1. DDS-DB Integration Mechanism

The DDS-DB integration mechanism contains a publication and subscription component with the four components being as seen in Figure 5.

DDS2DBMS Publication is a collection of functionality that disseminates topic instances as well as propagates the outgoing DDS samples to a DBMS table. DDS2DBMS Subscription is a collection of functionality that can propagate an incoming DDS sample to a table managed within the DBMS.

DBMS2DDS Publication is an enterprise application can change data within a DBMS table and ultimately have the table update information published, via a DDS DataWriter. DBMS2DDS Subscription will be associated with a table and will subscribe to associated DDS topics to capture.

4.2. Scenario of DDS-DB Integration

DDS2DBMS Publication/Subscription. When data in the DDS domain has to be available in the DBMS domain, the DDS-DB integration can be configured to facilitate that functionality. A topic in DDS will be mapped to a table in DBMS. The scenario is shown in Figure 6.

1. DDS application writes the temperature topics
2. Data is published using DDS and propagates the outgoing DDS samples to a DBMS table
3. DDS DataReaders receive data in same domain
4. DDS DataReaders store data in database
5. Data is accessible via SQL

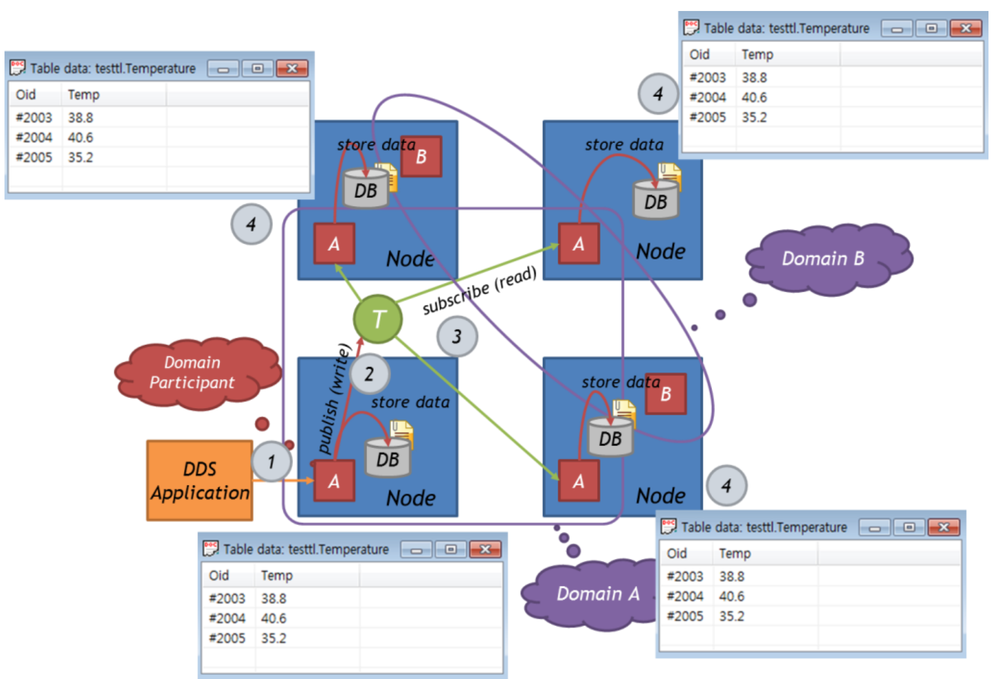

Figure 6. The Scenario of DDS2DBMS Publication/Subscription

DBMS2DDS Publication/Subscription. When data in the DBMS domain has to become available in the DDS domain, this can be achieved by configuring the DDS-DB integration to map a table to a topic. The scenario is shown in Figure 7.

1. Table is changed using SQL
2. DB-ReTiCom detects change and the data is published using DDS
3. DDS DataReaders receive data in same domain
4. DDS DataReaders store data in database
5. Data is accessible via SQL

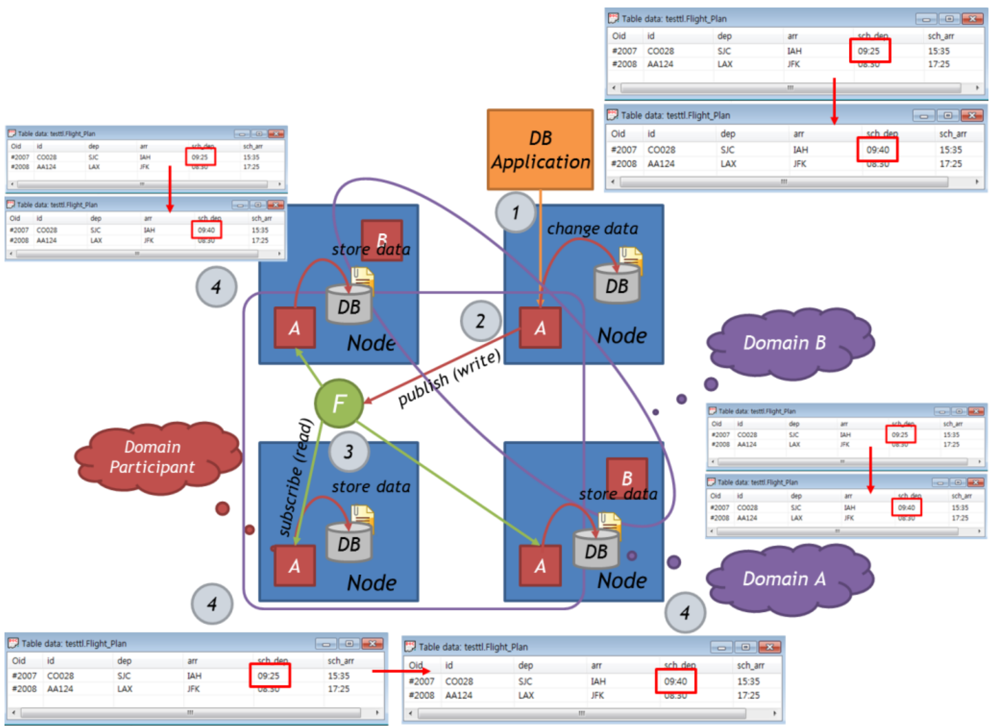

Figure 7. The Scenario of DBMS2DDS Publication/Subscription

5. Final Remarks

DDS-DB integration is an important issue with a growing interest. This paper introduced a mapping between DDS and DBMS meta-model and DDS-DB integration mechanism for supporting data persistence. The DDS-DB integration can bridge data from the DDS domain to the DBMS domain and vice versa. A topic in DDS will be mapped to a table in DBMS. When data in the DDS domain has to be available in the DBMS domain, the DDS-DB integration can be configured to facilitate that functionality. In the future, the authors intend to continue to work on bridge to incorporate real-time communication with persistence.

Acknowledgements

Authors are gratefully acknowledging the financial support by Defense Acquisition Program Administration and Agency for Defense Development under the contract UD100003KD.

References

[1] Object Management Group, "Data Distribution Service for Realtime Systems", v1.2, **(2007)**.
[2] Mark A. Hamilton, Real-Time Innovations, "Embedded to Enterprise Application Bridging (e2E) Utilizing DDS & RDBMS Technologies", **(2009)**.
[3] Object Management Group, "Common Object Request Broker Architecture (CORBA)", v3.0, **(2002)**.
[4] Ismael Etxeberria-Agiriano, Isidro Calvo, Federico Pérez and Oier García de Albeniz, "Mapping Different Communication Traffic over DDS in Industrial Environments", Information Systems and Technologies, **(2011)**.
[5] RTI DDS, http://www.rti.com/.
[6] OpenSplice, http://www.opensplice.com/.
[7] OpenDDS, http://www.opendds.org/.
[8] BEE DDS, http://www.beedds.com/.
[9] Twin Oaks CoreDX DDS, http://www.twinoakscomputing.com/coredx.
[10] K. -J. Kwon, C. -B. Park and H. Choi, "DDSS: A Communication Middleware based on the DDS for Mobile and Pervasive Systems", The 10th International Conference on Advanced Communication Technology, **(2008)**.
[11] FastDB, http://www.garret.ru/fastdb/FastDB.htm.
[12] Real-Time Innovations, "RTI Real-Time Connect for RTI Data Distribution Service User's Manual", v 4.5, **(2011)**.
[13] PrismTech, "OpenSplice DDS Development Guide", v 4.1, **(2009)**.

Authors

Hyung-Jun Yim is a Ph.D. Candidate in Chungnam National University, Daejeon, Korea. His current research interests are in the areas of Internet of Things, Cloud Connected Car and Data Distribution Service.

Yun-Hee Son is a Ph.D. Candidate in Chungnam National University, Daejeon, Korea. His current research interests are in the areas of Data Distribution Service and Linked Open Data.

Chan-Ho Song is a MS Candidate in Chungnam National University, Daejeon, Korea. His current research interests are in the areas of Internet of Things and Data Distribution Service.

Chumsu Kim is a senior researcher in Agency for Defense Development, Changwon, Korea. His current research interests are in the areas of computer engineering including distributed system, database and communication for naval combat management system development.

Gyoonjung Lee is a principal researcher in Agency for Defense Development, Changwon, Korea. His current research interests are in the areas of integration for naval combat management system and small target detection in sea environment.

Kyu-Chul Lee is a professor in the Dept. of Computer Engineering in Chungnam National University, Daejeon, Korea. He received his Ph.D. from Seoul National University in 1989. His current research interests are in the areas of XML, Web Services, Cloud Computing and Internet of Things. He has been responsible for several industrial projects and published numerous papers.