

## Developing an XML Configuration Tool for DDS

### ABSTRACT

In this paper we present our developed Tool for Automobile named Auto-Tool. An automobile has a growing number of signals after adopting X-by-wire systems. Our Project consists of developing an interface to manage automobile signals. These signals are provided by captors, actuators and central processing unit. Signals are transmitted over FlexRay bus taking into consideration the specific aspects of the publish/subscribe paradigm. We give an overview of works dealing with DDS and FlexRay. We present the role of the developed Tool to deal with Quality of Service of Data Distribution Service. Each DDS entity has specific parameters and these entities are dynamic. Entities are configured according to the Topic they are interested on thanks to Auto- Tool.

### Authors & Affiliation:

Wafa Najjar, Salem Hasnaoui

### Communications Systems

Research Laboratory,

SYSCOM Tunisa

Correspondence To:

Wafa Najjar

### Keywords:

Information System; Tool; DDS;

QoS ; Embedded System;

FlexRay

© 2014. The Authors.

Published under **Caribbean**

**Journal of Science and**

**Technology**

ISSN 0799-3757

<http://caribjscitech.com/>

## 1. INTRODUCTION

In many distributed embedded real-time applications, like military, automobile, aviation, telecommunications data produced in one component of the system needs to be shared with other components. The amount of information increases and much information needs to be real time.

In the last few years, x-by-wire in motor vehicles has been slowly working its way into production vehicles. The term "By-Wire" is the general term that means effectively replacing components such as the pumps, hoses, fluids, belts and vacuum servos and master cylinders with electronic sensors and actuators. X-by-wire technology in automotive industry represents the replacement of a pure mechanical system of motion or control with a sensor, an actuator, an Electronic Control Unit (ECU) and a communication network between them. In the automotive industry, X-by-wire technology is traditionally referred to primarily three categories: Throttle-by-wire, Brake-by-wire, and Steer-by-wire. These are listed in order of their perceived complexity and risk, from lowest to highest.

In order to satisfy needs in acquiring data at specific time and transmit them to concerned elements middleware was used. The middleware is a layer of software residing between hardware/network and applications/Services, providing a set of common and re-usable functions. These functions are accessible through APIs. Our research focuses on the middleware Data Distribution Service (DDS), which acts basic element of the system information. DDS is a Data-Centric middleware and it uses a global data space to exchange information. Our goal is to have a tool allowing us configuring DDS entities and abstracting programming complexity for the user. Auto-Tool gives the user the possibility of configuring system and imposing different quality of service parameter.

## 2. INFORMATION SYSTEM

An information system (IS) is a combination of information technology and people's activities that support operations, management and decision making. In a very broad sense, the term information system is frequently used to refer to the interaction between people, processes, data and technology. On one hand information system refer to the information and communication technology (ICT) that an organization uses. On the other hand, information system gives the way in which people interact with this technology in support of business processes.

Two concepts are strictly related to information system: computer system and business process. Information systems typically include an ICT component but are not purely concerned with ICT, focusing instead on the end use of information technology. Information systems are also different from business processes. Information systems help to control the performance of business processes.

An information system is considered as a special type of work system. A work system is a system in which humans and/or machines perform work using resources to produce specific products and/or services for customers. An information system is a work system whose activities are devoted to processing (capturing, transmitting, storing, retrieving, manipulating and displaying) information.

As such, information systems inter-relate with data systems on the one hand and activity systems on the other. Auto-Tool and DDS form an information system for the automobile. Auto-Tool is an API that abstracts the complexity of the middleware for the user. Auto-Tool is a part of an information system which supports human decision making and action. The middleware supports the decisions and configure the system as imposed by the user. Many middleware and API were developed in automobile domain; an overview will be given in section three.

## 3. DDS Middleware for information system

### 3.1. Basic DDS Feature

The challenge of an information system is to have a properly exchange of information. The suitable information must be given to the interested Entity at the right time without exceeding the delay. These requirements must be satisfied by a Soft element: the middleware. The middleware Achieve extremely low latency, high throughput and high availability for application-to-application communications. The middleware can reduce software complexity and accelerate development, integration and upgrades. This soft layer integrates disparate applications and systems. It can improve information sharing, responsiveness, and automation. Also the middleware can satisfy demanding real-time Quality of Service requirements.

DDS [1] is divided into two components: the Data-Centric Publish-Subscribe DCPS layer and the Data local Reconstruction Layer DLRL. The DCPS layer transports data from publishers to subscribers according to Quality of Service QoS constraints associated with the data topic, publisher, and subscriber. The DLRL allows distributed data to be shared by local objects located remotely from each other as if the data is local. The DLRL is built on top of the DCPS layer. DDS imposes the quality of service (QoS) in transmission between publishers and subscribers. There are many companies using the publish/subscribe paradigm in developing automobile middleware. An API is necessarily provided with the middleware to simplify management of information.

### 3.2. Quality of Service in DDS

The DDS relies [1, 2] on the use of QoS. A QoS is a set of characteristics that controls some aspects of the behaviour of the DDS service. QoS may be associated with all DDS entities such us Topic, DataWriter, DataReader, Publisher, Subscriber and DomainParticipant (Fig1). DDS automatically handles all aspects of message delivery, without requiring any intervention from the user application. DDS allows the user to specify QoS parameters [3] as a way to configure automatic- discovery mechanisms and specify the behaviour used when sending and receiving messages.

In DDS, the DataWriter gives samples to the Publisher who is responsible for taking the published data and disseminating it to relevant subscribers in the domain on a specific Topic. The subscriber receives the data from the publisher and passes it to any relevant data readers that are connected to it. Each DataReader is bound to a particular topic. This transmission could be named a distribution between publishers and subscribers.

In several cases, for communications to occur properly, a QoSPolicy on the publisher side must be compatible with a corresponding policy on the subscriber side. DDS will then determine whether the QoS parameters requested by the requesting side are compatible with what is offered by the offering side. If the two QoS Policy settings are not compatible, DDS will notify the offered side and communications will not be established.

Real time Qos parameters are: deadline, latency budget, lifespan, liveliness, reliability, presentation, ownership and durability. Deadline is an important parameter in hard and soft real time applications. Deadline indicates the minimum rate at which a Data Writer will send data. It also indicates how long a Data Reader is willing to wait for new data.

This policy is useful for cases where a Topic is expected to have each instance updated periodically. The default is a value of infinity; that is, there are no deadlines. Implementing deadline parameter gives many architectural Benefits:

- Allows each Data Writer to specify how fast it is able to publish data.
- Allows each Data Reader to specify how fast it needs to receive data.
- Indicates when a particular data source is not available, or that something may be wrong with a particular data source.
- Provides a way to set a time period during which a Data Reader will use the highest strength data source received when using multiple Data Writers.
- Allows an application to be notified immediately if a deadline is missed.

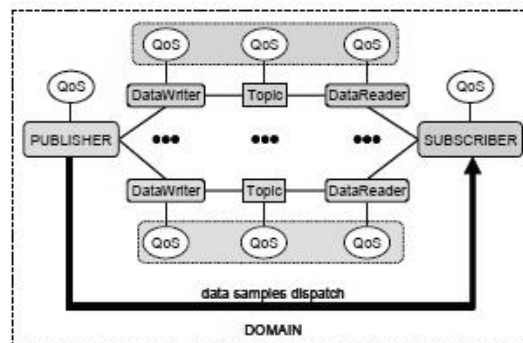


Fig .1. DCPS main entities and QoS Framework

All real time QoS concerns Topic, DataReader (DR) and DataWriter (DW), excluded Presentations QoS which concerns Publisher and subscriber. Specifications of these parameters is as following:

**Latency Budget:** specifies the maximum acceptable delay from the time the data is written until the data is inserted in the receiver's application-cache and the receiving application is notified of the fact.

**Ownership:** specifies whether it is allowed for multiple DW to write the same instance of the data and if so, how these modifications should be arbitrated.

**Reliability:** specifies whether or not a given DR will receive data "reliably" (no missed samples) from a DW. DDS will take care of the retransmissions of data if so required.

**Durability:** specifies whether DDS will make past samples of data available to newly joining Data Readers in the system

**Liveliness:** specifies how the DDS infrastructure determines whether or not a DDS entity is still "alive."

**Presentation:** specifies how the samples representing changes to data instances are presented to the subscribing application.

### 3.3 Overview of Existing DDS middlewares and DDS Tools

The Object Management Group gives the specification of the DDS middleware. Based on this specification, Middlewares are developed by many communities in order to perform the information system.

#### a) Open Splice

Open Splice DDS is a product developed by PrismTech. Prismtech is an acknowledged leader in performance-critical middleware and related services. The company develops markets and supports a wide range of high-performance, standards-compliant, and cost-effective middleware products which are further differentiated by their unique technical capabilities and advanced productivity tools. PrismTech's products enable event-driven communications, end-to-end Quality-of-Service, and the software optimization required to best-exploit the available computing resources to synchronize events, support system scalability and maximize system agility. Open Splice DDS includes a middleware, an API and a protocol for interoperability:

- Open Splice DDS DCPS is an implementation of Data Centric Publish Subscribe
- API Open Splice DDS DLRL gives the object model option
- Open Splice DDSI2 implements interoperable wire-protocol standard

Open Splice DDS component form the soft side of an information system. Open Splice DDS has become the most advanced, complete and widely used (commercial and Open Source) implementation of the OMG Data Distribution Service for Real-Time Systems (DDS) standard. Open Splice DDS enables seamless, timely, scalable and dependable data sharing between distributed applications and network connected devices. Its technical, operational, and financial benefits have propelled adoption across multiple industries. Open Splice DDS is:

- One of the fastest, most scalable and most reliable genuine open-source integration technologies
- Federated/Shared Memory application runtime deployment
- DDSI2: Proven interoperability with other DDS implementations
- A rich set of QoS policies for controlling efficiency, determinism and fault-tolerance

### b) Connex DDS

Real Time Innovations (RTI) implements the middleware Connex DDS. RTI provides the integration infrastructure for hundreds of real-time applications in industries such as defence, financial services and industrial control. RTI significantly reduces integration time, cost and risk with proven products and deep expertise in real-time systems. RTI is the world's leading provider of embedded middleware and of middleware that complies with the Object Management Group (OMG) Data

Distribution Service for Real-Time Systems standard (DDS). The company is a member of the OMG Board of Directors and has chaired the DDS working group since 2004.

There are five products in the RTI Connex family, all built on the Data Distribution Service (DDS) standard. These products are:

- **Connex DDS** is the world's leading implementation of the DDS standard, available in both commercial and free Open Community Source editions. It was previously referred to as NDDS and RTI Data Distribution Service.
- **Connex Micro** is a small-footprint DDS implementation for resource-constrained devices and applications that require certification to stringent safety standards.
- **Connex Messaging** includes Connex DDS and extends it with enhanced security, high availability and communication patterns.
- **Connex Integrator** mediates between applications that use different protocols or data models and federates systems across networks and security domains.
- **Connex Tools** offers a powerful set of debugging, testing and tuning capabilities for systems that use RTI Connex and the DDS standard.

### 4. Automobile Configuration for FlexRay

A Benchmark, based on Society Automotive Engineers (SAE) Benchmark, was presented [6]. The author has added to the original benchmark a number of nodes and messages to better represent the complexity of today's vehicles and to model some added options responsible for improving vehicle safety, reliability, cost and luxury. This Benchmark was designed to best fit the CAN network and was modified in [4] to be adapted to the FlexRay protocol. The resulting architecture is composed of 15 nodes connected by FlexRay bus. According to the FlexRay specification, each node consists of a Central Processing Unit that processes incoming messages and generates outgoing messages, a communication controller (CC) that independently implements the FlexRay protocol services, and a two-way controller-host interface (CHI) that serves as a buffer between the host and the CC. These nodes send 36 messages among which there are 30 periodic messages that need to be scheduled on the FlexRay static segment. The remaining six messages are sporadic and needs to be mapped into the dynamic segment.

### 5. Composition of Information System

An Information System encloses software and hardware elements us mention in figure 2.

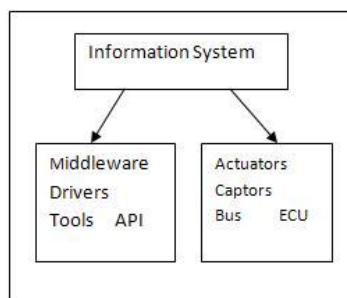


Fig.2.Information System

In 2008, our research team has projected to implement a DDS middleware satisfying Quality of services, independent from the environment. DDS is a standard designed to address the requirements of large-scale, mission-critical and real-time systems.

The project includes three subprojects. The first subproject consists of developing the middleware, the second one consists of analyzing the transmission over FlexRay and calculating the deadline for automobile signals and the third one is creating a tool to manage DDS entities and specifying the quality of service.

### 5.1. Middleware Implementation

We have considered the Object Management Group specification. The specification is given in an 'idl' file. We choose to implement DDS in the C language. Our methodology consists of using mapping rules idl to C in order to have the middleware structure. The C language was used because it is an embeddable language. The transmission bus considered is FlexRay. FlexRay driver and DDS are implemented under a real time operating system the  $\mu$  C-OSII. We have proposed the implementation of DDS Listeners over FlexRay [7].

### 5.2. Automotive Signals Transmission over FlexRay

In DDS, the DataWriter gives samples to the Publisher who is responsible for taking the published data and disseminating it to relevant subscribers in the domain on a specific Topic. The subscriber receives the data from the publisher and passes it to any relevant DataReaders that are connected to it. Each DataReader bound to a particular topic. This transmission is assured by the FlexRay. The FlexRay [5] bus gives transmission over static and dynamic bus and permits event triggered and time triggered transmission [9,10]. Many researches focus their work on deadline which is a real time QoS parameter [11]. We are also interested in deadline and we choose the automotive domain. The signals of automobile are specified in a Benchmark. In our work we deal with the Society Automotive Engineers (SAE) Benchmark signals. SAE [4] classified the automotive communications protocols into three classes A, B, C and D according to the message rate. Multiple signals were combining in a single message to reduce the effect of protocol overhead. The resulting message structure contained 17 messages [8].

### 5.3. Automobile Tool

We develop an Application Programmable Interface in order to abstract the middleware layer for the user and to give him the opportunity to modify DDS entities quality of service. We classify SAE Benchmark signals according to the publish Subscribe model. We identify DataReaders and DataWriters node. We classify all topics interesting Publishers and Subscribers. The characteristics of our Tool are:

- Abstracting the middleware layer
- DDS Nodes Configuration is possible without programming
- The user make decision through an API
- Parameter modification is simple
- Control of distributed system
- Adding entities promote system scalability

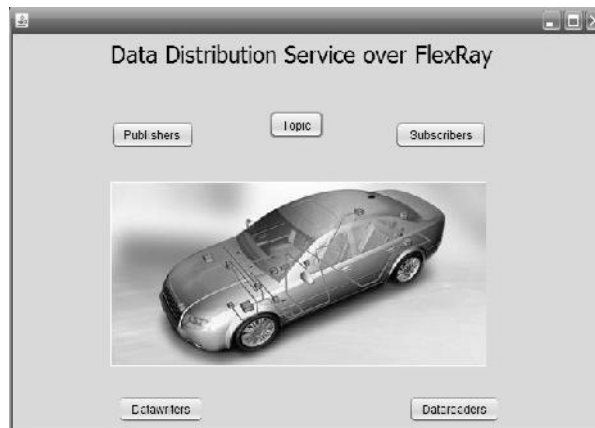
## 6. Automobile Information System Tool Implementation:

### 6.1. Software :

Automobile Tool was developed using NetBeans. Netbeans is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages. NetBeans is also referred to as a platform of modular components used for developing Java desktop applications. NetBeans IDE provides first-class comprehensive support for the newest Java technologies and latest Java enhancements before other IDEs. It is the first IDE providing support for JDK 7 (Java Development Kit) and Java EE 6. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, NetBeans IDE sets the standard for developing with cutting edge technologies out of the box.

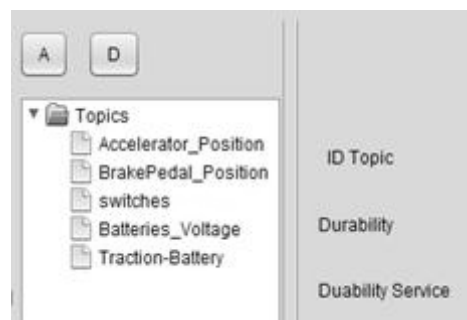
**6.2. Interface:**

The interface developed deals with all DDS entities. For that the main interface gives the user the access for the Publisher, the Topic, the Subscriber, the Datareader or the Datawriter displayed in figure 3.



*Fig.3. Main Interface*

When choosing the Topic Button, we have a tree enumerating all Topics. We can specify QoS parameter such Durability mentioned in figure 4.



*Fig 4. Topic Interface*

**6.3. Link between AUTO-Tool and DDS middleware:**

The AUTO-Tool is communicating with the middleware through XML files. We implement parsers to update DDS entities information and to guarantee Quality of Service configuration. The middleware has to apply user specification by extracting information corresponding to each entity.

**7. CONCLUSION**

In this paper, the development of an Automobile Information System tool was presented. Auto-Tool allows to the user the configuration of DDS entities and the specification of entity quality of service. The tool is developed in Java and uses xml files to save entities information and ensure the link with the middleware.

In future, it is still desirable to upgrade Auto-Tool and giving to the engineer the possibility of generating software components

**REFERENCES**

1. Object Management Group, "Data Distribution Service for Real-time systems", January 2007.
2. Prismtech "Open Splice C reference Guide "version 2.2, December 2006
3. M.Ryll, S.Ratchev,"Towards a publish/subscribe control architecture for precision assembly with the data distribution service",2004.
4. H.Jaouani and al, "Hybrid Task and Message Scheduling in Hard Real Time Distributed Systems over FlexRay bus", *International Communication and information Technology*3, p21-26, 2012.
5. FlexRay Communication System Protocol Specification Version 2.1, 2004-2005
6. U.Mohammad, N. Al-Holou,"Development of An Automotive Communication Benchmark", *Canadian Journal on Electrical and Electronics Engineering* Vol.1,No.5, august 2010.
7. R.Bouhouch, W.Najjar, H.Jaouani, S.Hasnaoui, "Implementation of Data Distribution Service Listeners on Top of FlexRay Driver", *INFOCOMP 2011 : The First International Conference on Advanced Communications and Computation, IARIA*, 64-69
8. W. Najjar, H. Jaouani, R. Bouhouch and S. Hasnaoui, *Analyzing Real Time Communications on DDS over FlexRay*, *IEEE LATIN AMERICA TRANSACTIONS*, VOL. 10, NO. 1, p.1149- JAN.2012
9. J.Hillebrand, M.Rahmani,R.Bogenberger,"coexistence of Time-Triggered and event triggered traffic in switched full-duplex Ethernet Networks".
10. T.Pop,P.Pop,P.Eles,Z.Peng,"Timing analysis of the FlexRay communication protocol", Springer Science, October 2007.
11. B.Kao,H.Gracia-Molina ;"Deadline Assignment in a Distributed Soft Real-Time System", *IEEE*, December 1997