

Integration of Qt Based UI with DDS Middle Ware for Controlling of QoS Policies for Real Time Warfare System

Akif Naz¹ Gambhir Singh²

¹M.Tech Scholar ²Associate Professor

^{1,2}Department of Computer Science Engineering

^{1,2}HR. Institute of Technology, Ghaziabad, U.P.

Abstract— In the current age technologies has become more advanced and robust. The implementation of this advanced concepts should be efficiently incorporate for security purposes. For the security of country our real time automatic command and control warfare system should use these advanced concept in there operation. Keeping this view in mind, present study focused to develop an integrated model .This Model has implemented effective use of a middle ware technology as Data Distribution Services (DDS) with unique feature of Qt as signal and slots mechanism and User friendly user interface . Qt Based User Interface with Data Distribution Services were integrated in this model for Applying & Controlling QoS at run time using user friendly Interface, and also the evaluation of conventional Socket communication is referred to DDS which lead us to the outcome that DDS is much more efficient for real time critical system like war fare system , industrial automation etc.

Keywords: DDS, Warfare, Qt Creator, QoS, Qt GUI

I. INTRODUCTION

Object Management Group's Data Distribution Services (DDS)[1] is an open platform independent middleware Solution for data Distribution using Publisher Subscriber model[1][2].DDS infrastructure solution contains many quality of Services (QoS), by applying these DDS QoS the behavior of data distribution and transmission can be controlled[1].

On the other hand Qt is an open C++ based User Interface(UI) tool kit and framework for rich and user friendly (Graphical User Interface) GUI by the Qt Company [3]. Qt is a cross-platform application framework with rich graphical user interface that is being used for mobile and application software development [3][4]. It is currently active by Qt Company , Digia and previously by Nokia , initially it was by the Norwegian company Trolltech , the original producer of Qt [3][4] . In the era of advanced technology for the real time automatic command and control system should robust with more efficient architecture and dynamic flexibility. To developed a better real time warfare system, there is a need to take proper steps, as first to compare the used technologies, finding out best of it and integrate technologies to develop a effective functional running system.

II. LITERATURE REVIEW

Existing middle ware technology which is being used for data passing such as Java RMI , Com , CORBA , JMI and Web Services are being used to fulfill the demand of a System with Complicated and heavy process Mechanism for communication with a lack of run time flexibility and Quality of services For the middle ware data communication there are many data Publisher/Subscriber system available but these

systems are specific in nature, no generality in existing system , to over come these issues OMG Developed a generic model called DDS an open and platform-independent QoS based middleware standard that uses the refined concept of Publish/Subscribe[1][2]. Accuracy and time bound communication is necessary in data communication for the software's of Defense sector medical etc. DDS Middleware system is basically open standard for data distribution for real time communication with high accuracy. DDS Characteristics are itself able to solve many problem of business or a tactical command information system which is hard to understand and solve in these system[5]. For the middle ware many technologies are being used such as CORBA ,COM, Web Services , Java RMI etc. which help to increase efficiency and reduce cost of the system [6] and it is necessary to improve middle-ware for the improvement of complex system. For data transmission from one node to other many middleware standard and protocols are being used. Different protocols have their different set of rules for data transmission. The basic protocol are UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) [7]. User Datagram protocol is unrealizable connection less protocol , which does not establish connection between sender and receiver before data sending , and also sender does not acknowledge about the date sent by sender . UDP is consider where reliability is less important than time delivery of data [7][8] .Transmission Control Protocol is the connection-oriented, reliable protocol which establish connection between sender and receiver before sending data and receiving end also send acknowledgment to the sender[7][8]. These protocols have their own advantages and disadvantages if used to send and receive data. Considering these protocols OMG Developed Data Distribution rules called DDS as Data Distribution Service as an open and platform independent middle-ware standard with minimal overhead[9] . This is an open standard for many to many communication and also provide various quality of services for data controlling during communication.

A. Selection of Graphical User Interface (GUI):

Before developing a software , there are no of factors taken into account which language or technology is the best for a particular product .These factors may be as how efficient a language for programmer, how efficient for memory management, how efficient a language at run time .

Here we are considering two approaches for Graphical User Interface, integration with DDS for QoS Control. In Java Unused free memory is automatic handled by java run time system[10].Programmer need not to write code for handling memory. Java use garbage collection feature. Java Virtual Machine (JVM) automatically checks all the allocated memory blocks in a period , if any block of memory has no reference, removes that memory block. But it

add extra overhead which may lead to extra memory consumption and slow runtime speed[10].

In other language like C++, the programmer has to write code to delete blocks of memory if memory is no longer required. With Java garbage collection running depends on JVM implementation to delete occupied memory. Because of the periodically checking of memory block ,it also hamper the run time efficiency of a java program[10].

C++ compiler compile C++ programs directly into a binary format and this binary or machine code directly execute on hardware. But in Java , Java Code first Converted into intermediate code which is called byte code, and this byte code again executed by JVM(Java Virtual Machine)to run Java program. So this kind of execution on hardware is compare ably slow. Just in Time compilers can also be used for that but this is not also a universal solution [10]

As the ultimate goal to integrate of DDS with GUI for QoS Controlling So Before creating a GUI, this is very much necessary to create GUI which is efficient in all manner. So here comparison of GUI Java UI like AWT and Swing, with the C++ based cross-platform toolkit Qt, initially supplied by Trolltech .There are two technique of UI Development in Java first AWT (Abstract Windowing Toolkit).It uses native code for the GUI components and give some wrapper code for GUI. So Because of the native code is different for different machine so when a AWT GUI run it look and behave differently on different platform. Advanced side java use Swing. It uses native libraries only for very basic things and other things handled by Swing itself but there are some efficiency issue. Swing programs are slow and give poor response with modern hardware [10] . Similarly, C++ based Qt toolkit follows a similar approach like Swing, it only relies on the native libraries only for very basic things and handles the drawing of GUI components itself. Its look and feel remain same for all the platform as swing and as Qt work with C++ thus compiled to native code; it does not have Swing's efficiency problems[10]. Because of caching techniques of Qt All User Interface are very fast as compare to native APIs based approaches. With GUI , event handling mechanism is always work parallel which can be easily handled with the Qt SIGNAL and SLOT feature[3] as compare to the Java Event handling classes.

III. PROPOSED MODEL

Proposed Model is an integrated model of Qt based User Interface with middle ware Publisher Subscriber DDS model (Figure 1) and selection of Quality of services(DDS QoS) at run time by using user interface for reducing time and increasing efficiency of a tactical warfare system. For the development and integration used software are RTI DDS APIs[11], Qt 4.8 , Qt Creator 2.4.1[3][4] , Linux Os ,Wire shark , Ms Excel etc .

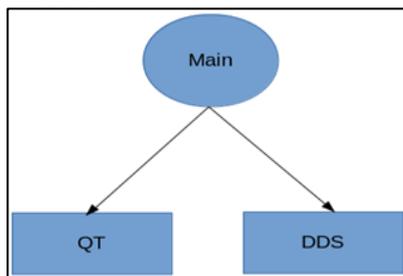


Fig. 1: Shows Basic Integration Model of Qt and DDS

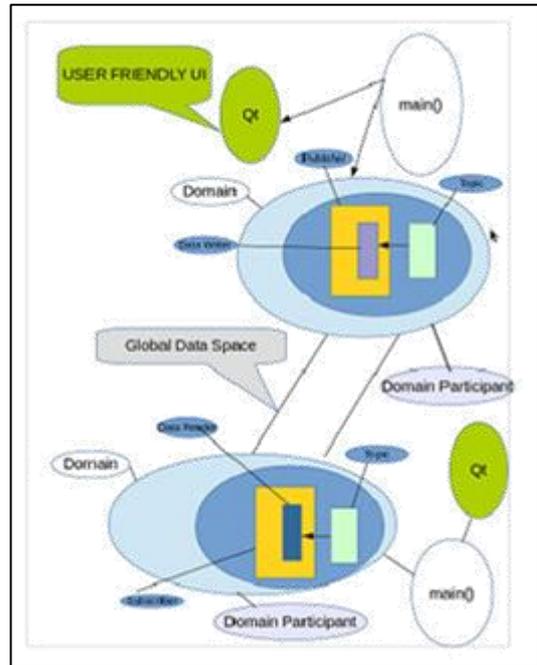


Fig. 2: Integrated Model For Data Communication Using Qt & DDS.

A. DDS Architecture:

In the Concept of DDS Data exchange , DDS provide a global data bus in which publisher and subscriber read and write data[1]. In DDS, Domain is the Global Data space and every domain is identify an integer number[11] . To excess the domain, Domain participant is used .One domain data is not accessible to other domain but an application can use multiple domain by creating one domain participant for one domain[11]. Domain participant is a facility to use domain. It serve as a factory, container and manager for DDS entities[12].

B. Publisher and Data Writer:

To write data, there is a need of a publisher and a data writer . A publisher is created and attached with a domain participant and used as a factory to create and manage a group of data writers[12] these publisher publish their data in the same logical partition within the global data space[12]. Publisher and Data writers have related operation and both have similar Quality of services, on the basis of these QoS the work of data writing complete.

RTI DDS has given APIs to send data from one node to others and visa versa. The main aim of this study to explain DDS functions call using RTI and Qt feature for underrating, as Topic is a kind of packet which has a name , type and services. To Send data from publisher to subscriber first has to define a structure and fill data to the structure , structure to send data from one node to another.

Here used of DDS APIs is given by the DDS RTI [11] [13] .

C. Subscriber and data Reader:

Same as data writer, there is need of data reader and subscriber. To read data, subscriber and a data reader are created . A Subscriber is created and attached with a domain participant and these subscriber subscribe data in the same logical partition within the global data space. Subscriber and Data reader have related operation and both have similar

Quality of services, on the basis of these QoS the work of data writing is complete

D. Sending Side Steps:

- 1) Create Process
- 2) Create Ui Thread
- 3) Create Com Dds Thread
- 4) Create_Participant(Id, ,Null, Mask);
- 5) Create_Publisher(Qos, Null, Mask);
- 6) Create_Datawrite(Topic,Qos,Null,Mask);
- 7) Create_Topic("Name",Type, Qos,Null,Mask);
- 8) Select Qos By Ui
- 9) Set Qos In Dds
- 10) Send Data .

E. Receiving Side Steps:

- 1) Create Process
- 2) Create Ui Thread
- 3) Create Com Dds Thread
- 4) Create_Participant(Id, ,Null, Mask)
- 5) Create_Subscriber(Qos, Null, Mask)
- 6) Create_Datareader(Topic,Qos,Listener,Mask)
- 7) Create_Topic("Name",Type, Qos,Null,Mask)
- 8) Select Qos By Ui
- 9) Set Qos In Dds
- 10) Receive Data .

1) DDS_DomainParticipantFactoryQos factory_qos;

Apply QoS, QoS basically facility or services to send data from one node to other ,these sevice can be applied at different stages as per requirement.

get_qos() , set_qos() [11]

First of all create a text file with *.idl extension, In this file create data struture that will need to send from publisher to subscriber .as structdata.idl

Now Need to Use RTI InBuild tool to generate Supporting code for communication. Use command as-rtiddsgen -language C -example i86Linux2.6gcc4.4.5 structdata.idl. In this command rtiddsgen is a tool , generated code in c and code for Linux 64 bits.The generated code by rtiddsgen comes in many categories as , DDS Data carrier source code for data communication and controlling thread and code for various state of data. Here Xml file for applying QoS policies Compile generated code and Run publisher subscriber code[11].

There are several approaches to load QoS profile from xml files. Creating a xml file , which loaded automatically with default QOS Services and on the other hand load QOS services with customized services and added in Domain Participate Factory[11]. In our solution we are selecting QoS policy on run time on the basis of Qt based smart UI.

In this paper the complete system is divided in to two parts. One part is called Static Model and other part is Dynamic Model, Static model is a manual model. In QoS Based DDS project xml file defines Quality of Service (QoS) and these QoS policy parameter decide the behavior of communication. In Static model we are taking those project in which xml are directly connected to the communication code, So to change communication behavior of these model underline QoS policies has to be changed and for this , directly connected xml file parameters need to be changed.

In the Dynamic model the defined QoS policies can be easily handled by a integrated Qt based Graphical user

interface , these Qt based UI is directly connected to xml for QoS Policy. To Change the behavior of Communication, QoS Policy can be easily changed by connected UI.

F. Benefits of Dynamic model as compare to Static model are given below:

1) Loosely Coupled:

Dynamic model is a loosely coupled model,in which used Quality of services can be easily changed by the mean of connected graphical user interface, and in the case of static model user has to changed xml for different QoS policy.

2) Flexibility:

In dynamic model changing policy is very flexible by using user interface as compare to xml files in static model.

3) Error Free:

There is no chance of error in dynamic model, to change policy by using GUI is error free, but in case of static model where user has to modify xml files changing policy is very tedious or error prone.

4) Delay:

There might be a scenario of communication in war fare system where at a point of time some data is required to be send to multiple nodes or to some specific node, these kind of behaviors Or even more complex communication behavior can be controlled by changing Quality of services (QoS) , the delay time taken by dynamic mode is comparatively less than static model.

IV. RESULTS & ANALYSIS

In the real time warfare software system where many systems are involved these systems need to communicate to each other in a real time with high accuracy and best precision . This kind of system also requires information security, previous data history feature , no missing of packets and others. Old system was not able to provide that kind of facility as provided by DDS and also In point to point socket communication each node has to connect to each other using IP address of others and also has to bind using previously assigned port number, this kind of structure creates complexity for communication when hundreds of nodes are being used for communication. In addition DDS communication every node has to connect to a central mechanism and transfer data using topics. Integration of DDS with Qt feature makes it more powerful.

Using Xml Changes (Manually)		Using User Interface(Qt Gui)	
Policy Applying Time O'clock	Policy Start Time O'clock	Policy Applying Time	Policy Start Time
1.0	1.7	3.0	3.02
1.20	1.27	3.04	3.06
1.30	1.35	3.07	3.08
1.40	1.45	3.10	3.10
1.45	1.49	3.14	3.15
1.50	1.55	3.16	3.17
1.56	2.00	3.19	3.21
2.05	2.07	3.24	3.24
2.08	2.13	3.27	3.29
2.14	2.17	3.31	3.31
2.18	2.21	3.33	3.34
2.23	2.27	3.35	3.36

2.28	2.31	3.38	3.39
2.33	2.36	3.41	3.42
2.40	2.43	3.44	3.44
2.45	2.49	3.46	3.47
2.52	2.56	3.49	3.50
2.57	2.59	3.52	3.52

Table 1: Time taken to change QOS by using xml v/s. using Qt GUI.

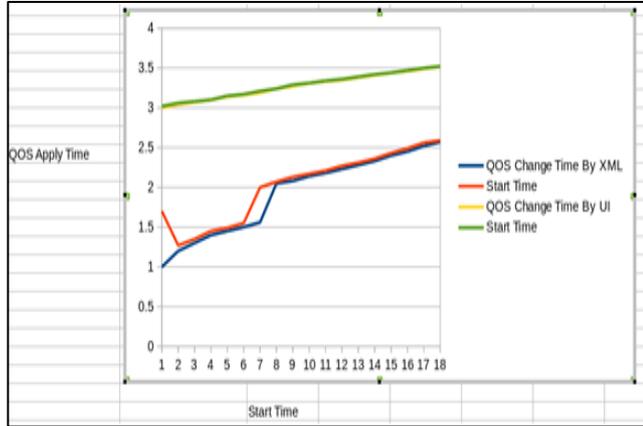


Fig. 3: Shows Change in QOS by Manual XML Vs Qt UI

To measure the attributes value of socket and DDS, like Round trip time (RTT) and Latency, test has been performed. Given RTT value is the average of 5.00.00 iterations and after the evaluation of RTT, latency can be calculated on the basis of given formula.

$$\text{Latency} = \text{Round Trip Time} / 2$$

Given below table shows differences in the attributes values for DDS and Socket

Data(Byte)		16 Byte	32 Byte	64 Byte	128 Byte	256 Byte	512 Byte	1024 Byte
Socket(us)		15	15	15	15	16	16	17
DDS(us)	Volatile QoS	15	15	15	15	16	16	17
	Transit QoS	16	16	16	16	17	17	18

Table 2: For Measuring RTT in socket v/s DDS

Data(Byte)		16 Byte	32 Byte	64 Byte	128 Byte	256 Byte	512 Byte	1024 Byte
Socket(us)		75.5	76.5	79	79	80	84	86
DDS(us)	Volatile QoS	76	76.5	78.5	79	80.5	83.5	86.5
	Transit QoS	81	82.5	83	83	86	89	91.5

Table 3: For measuring Latency in socket v/s DDS.

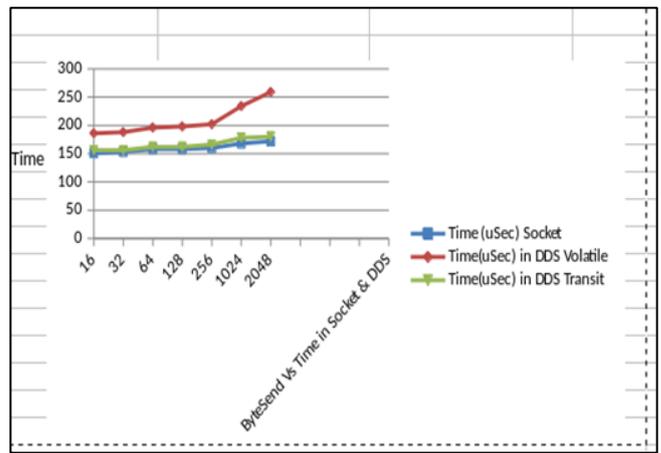


Fig. 4:

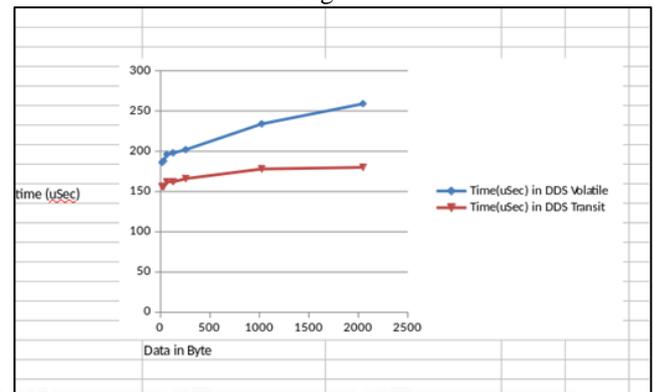


Fig. 5:

Data Availability for Late comers: In case of socket communication there is no provision of data availability for late comers, if a node join in communication after data sent, node not able to receive data. But in the case of DDS, due to the facility of QoS [1], late comer node can also get data, which has been sent earlier. Although round trip time is little bit increase in case of Transit QoS due to the data saving feature, which provide saved data for the late comers node. Below table given for better understanding.

QoS Effect	Data Availability after Transmission		
	Volatile	Transit	
Socket	---	---	Data not available for latecomer node
DDS	Not	Yes	Data available for late comer node.

Table 4:

Hence as a result we see that DDS includes lot of features like high accuracy No loss of packets, Data availability for late comers, as compare to socket based warfare system. Therefore, Integration of DDS with Qt GUI and signal slot feature makes it more time effective and user friendly for taking timely decision by controlling QoS at run time for heavy command and control systems like warfare system.

V. CONCLUSION

Here there is a comparison of the two development technology Java(AWT/Swing) and C++(Qt) regarding their efficiency and suitability developing Quick, high performance, user-friendly applications with(GUI) graphical

user interfaces Or even Qt does not force for particular programming paradigms as Swing does with the Model-View-Controller(MVC) paradigm. Qt Developer often get more concise code. Qt library for more Quick response user interface and signal slots mechanism make it powerful to integrate with highly efficient data distribution communication system. Real time war far system where timely delivery of message , message history, security of messages and quick response of messages is required DDS and Qt base system is the best system .In which Qt provide User Interface and DDS provide communication facility .

Real time war fare system always require a reliable and accurate system to communicate information during war time .For fulfilling of this requirement DDS has inbuilt features of QoS , and Integration of it with Qt UI makes it more powerful and User Friendly.

REFERENCES

- [1] Object Management Group, OMG. Data Distribution Service for Real-time Systems specification, version 1.2, 2007.
<http://www.omg.org/spec/DDS/1.2/http://portals.omg.org/dds/>
- [2] B.Almadani, M.Naseer Bajwa, S.Yang, A.Saif, "Performance Evaluation of DDS Based Middleware over wireless Channel for Reconfigurable Manufacturing System International journal of distributed sensor networks"Article ID 863123, Vol 2015 ,11 pages<http://dx.doi.org/10.1155/2015/863123>
- [3] Wiki.qt.io , <https://wiki.qt.io>
- [4] N.SaiJithendra, D.Aruna kumari, K. V. Murali Mohan "The Research of Qt_Embedded and Embedded Linux Application in the Intelligent Monitoring System Control", www.ijser.org
- [5] X.Ma, S.Liu,"Research on General Data Publish/Subscriber Model Base on DDS" Trans Tech publisher, Switzerland ,2013 , ISSN ;1662-7482, Vol 397-400 pp 2491-2494.
- [6] B. Almadani , A.A.Roubaiey, R.Ahmed "Manufacturing System Integration using Real Time Qos-Aware Middleware"Trans Tech publisher, Switzerland ,2013 ISSN ;1662-8985, Vol 711 , pp 629-635.
- [7] https://en.wikipedia.org/wiki/User_Datagram_Protocol
https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [8] S.Kumar, S.Rai "Survey on Transport Layer Protocols: TCP & UDP" , International Journal of Computer Applications (0975 – 8887) Vol 46– No.7, May 2012
- [9] H.Bian , Y.Zhuang, D.Chen " Research On DDS for Aircraft Collaboration Design System" Communications in control science and Engineering, Vol - 2 , 2014 www.as-se.org/ccse.
- [10]M. Kalle Dalheimer , K.Datakonsult AB, "A Comparison of Qt and Java for Large-Scale, Industrial-Strength GUI Development"
<http://ftp.icm.edu.pl/packages/qt.old/pdf/qt-vs-java-whitepaper.pdf>
- [11]<https://community.rti.com>.
- [12]R. Bouhouch, W. Najjar, H. Jaouani, S. Hasnaoui,"Implementation of Data Distribution Service Listeners on Top of FlexRay Driver"
INFOCOMP 2011 Tunisia, The First International Conference on Advanced Communications and Computation.https://www.researchgate.net/publication/268422625_Implementation_of_Data_Distribution_Service_Listeners_on_Top_of_FlexRay_Driver
- [13] Wikipedia RTI DDS.