CONTROLS MIDDLEWARE FOR FAIR

V. Rapp, GSI, Darmstadt, Germany
W. Sliwinski, CERN, Geneva, Switzerland

Abstract

With the FAIR complex, the control systems at GSI will face new scalability challenges due to significant amount of new hardware coming with the new facility. Although, the old systems have proven themselves as sustainable and reliable, they are based on technologies, which have become obsolete years ago. During the FAIR construction time and the associated shutdown GSI will replace multiple components of the control system. The success in the integration of CERNs FESA and LSA frameworks had moved GSI to extend the cooperation with the controls middleware and especially Remote Device Access (RDA) and Java API for Parameter Control (JAPC) frameworks. However, the current version of RDA is based on CORBA technology, which itself, can be considered obsolete. Consequently, it will be replaced by a newer version (RDA3), which will be based on ZeroMQ, and will offer a new improved API based on the experience from previous usage. The collaboration between GSI and CERN shows that new RDA is capable to comply with requirements of both environments. In this paper we present general architecture of the new RDA and depict its integration in the GSI control system.

INTRODUCTION

Officially started in 2010 the FAIR (Facility for Antiproton and Ion research) [1] project will extend the existing GSI accelerator by many new installations. Those will introduce a plenty of new hardware, which need to be managed and monitored by the control system.

To benefit from the existing experience GSI has decided to start collaborating with CERN on the device software framework called FESA. This collaboration was successful, so the future devices, which will be introduced within the scope of FAIR will be developed with FESA framework. The FESA framework itself is designed to operate within the CERNs system context and depends on the underlying CERN middleware and the messaging layer called Remote Device Access (RDA). The adaptation of the FESA framework to the current GSI middleware is difficult. Moreover the current version of the GSI control system depends on CORBA [2] which itself is a rather an old messaging library with lack of support and shrinking community. Additionally this year CERN released a new RDA version based on the ZeroMQ [3] messaging framework, which is light, modern and well suited to operate in accelerator environment [4]. Therefore the further devices developed for the FAIR project will utilize the new middleware solution.

On the other hand GSI itself is running a considerable amount of old hardware. This is done by using of the own developed middleware messaging layer called Device Access. Upgrading of all old hardware will be costly and difficult to accomplish. Hence the middleware architecture of FAIR will consist of two different parts: the CORBA based Device Access and the new ZeroMQ based RDA.

DEVICE ACCESS

Today's control system at GSI can be roughly divided into 3 parts. First the real time processes run on so called Front End Computers (FEC), which control the equipment and collect the information from it. On the other point of the architecture the mostly Java based Graphical User Interface (GUI) applications, which present the gathered information and allow operators in the control center to set hardware parameters of the beam equipment. Additionally the information form FECs is also monitored and collected by different automated services like logging or monitoring. Those two parts are connected by a middleware called Device Access.

Each process running on FEC is modelling a particular device of the accelerator equipment which is physically connected to FEC and managed by it. A typical device model is an object oriented device presentation, which consist of multiple readable and writable properties (e.g. device status, voltage, measured pressure etc.). Each model can also offer particular methods to control the connected device such as "reset" or "initialize", which can be called from other tiers using the middleware. To access the properties from another process like GUI, Device Access provides basic get, set and method call mechanisms, but also allows subscription for particular properties. Device Access also supports a given context for property values, which is used in multiplexed beam operation. To distinguish beams a simple numeric scheme is used. A basic number is associated with particular beam. Those numbers are referred as virtual accelerator.

Device Access allows addressing by device names. Those are rather convenient to users then the network addresses or CORBA IORs. It also provides flexibility to change the network address of particular FEC, without the need of changing the user software. The name resolution is done by the naming server, which stores the CORBA IORs of particular FECs assigned to their device names. The name resolution itself is transparent for the user and is performed by the middleware.

In addition to functions described above Device Access also provides a basis access right control mechanism [5]. This function is provided by the same server as the name resolution.

ISBN 978-3-95450-146-5

THE CERN CONTROL SYSTEM

The current control system at CERN is implemented in a similar way as the one at GSI. It is originally based on CORBA and operates with a similar device-property model. However with over 4000 servers [6] it handles a much larger amount of devices then its counterpart at GSI. Additionally the CERNs heterogeneous infrastructure with a lot of legacy equipment and operation systems is facing a bigger compatibility and management issues. Which means however, that it is proven to be flexible and performant enough to operate in environments like FAIR. Due of the scale and restrictive requirements considering logging and fault tolerance CERN is also running additional messaging systems on top control middleware [6].

CERN middleware is using its own developed role based access system called RBAC [7], which bring its own infrastructure and hence the only focus of the naming server (called Directory Server is CERNs context) is device name resolution.

RDA

One of the core parts of the CERNs control system is its messaging layer called Remote Device Access (RDA). Originally RDA was based on a CORBA technology, which can be considered obsolete. As many other CERN libraries RDA is provided natively for both C++ and Java platforms. C++ part of RDA depends on the OmniORB [8] library, while the Java part uses the JacORB [9]. In 2011 CERN have started a project to replace the underlying CORBA layer by modern software. After the evaluation of possible candidates to replace CORBA, ZeroMQ was chosen as the most suitable messaging framework for the Control System tasks [4]. In July 2014 a new version of the device access middleware was released - RDA3. Although, sharing the same concept it is a completely new development based on ZeroMQ. It introduced a new API, which is not compatible to the old one. This was done mainly to simplify the new interface and remove old ballast. The compatibility on the API level however can be achieved by using of additional proxy translation services, which are also a part of the middleware.

Similar to previous version RDA3 is operating on the same device-property model and supports get, set and subscribe operations on defined properties. Same as RDA2 both C++ and Java versions are provided. They share a similar API with same class, method and property names. Both versions are independent from each other and can be used natively in according environment.

RDA uses the same device property abstraction level allowing the users an easy and transparent access to the properties. The naming resolution is achieved by the same service called Directory Server as in previous version. To manage this, the Directory server is keeping track of all deployed devices. Therefore each running RDA server is registering himself on Directory Server and providing him with the necessary connection information (see Fig. 1). The server is compatible with both RDA2 and RDA3 and uses the same data structure to keep the information. Beside of the servers name the connection information contain a CORBA IOR In case of the RDA2 or a specific connection identifier string in case of RDA3. Using this data client part of the middleware is able to communicate with a desired server. The information about the particular devices deployed on the server in not transmitted in this step, since the implementation of the device model itself is done in the upper layer. To provide the connection information for a particular device the server is accessing a database, which contains the information about all installed device software and in particular on which RDA3 server those devises are deployed. This database is filled during the device deployment processes.

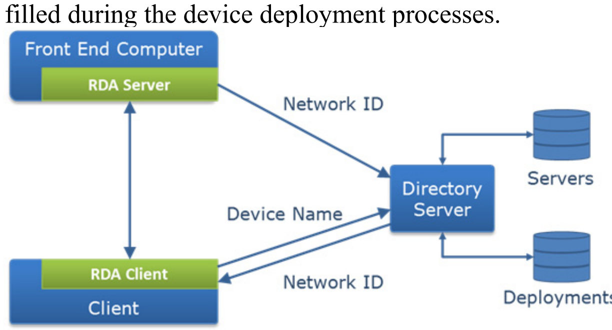

Figure 1: Communication with Directory Server.

Obviously the full and immediate replacement of old middleware by the new version cannot be done on the running system. Hence CERN will replace the old parts step by step. Some parts are already running with RDA3 and show good performance and user satisfaction.

FAIR MIDDLEWARE

As mentioned before the further device software developed for the FAIR project will be using the RDA framework in its laatest version. In 2013 GSI has extended the collaboration with CERN by the controls middleware and especially RDA framework and is successfully contributing to the software. Beside of the knowledge exchange, this collaboration also allows GSI to implement own particular requirements.

The big amount of existing devices currently operated with the old Device Access software makes the migration to a new framework a difficult and costly task. Hence at the beginning of the FAIR operation both messaging layers will be present in the control system software landscape. Facing same issues on its side, CERN it has developed a transparent Java API called Java API for Parameter Control (JAPC). This API acts as an additional layer and hides the underlying middleware implementation from the user. It makes a middleware-unaware access possible. Most of the top level application is written in Java, so this API is developed only for this platform.

To access a particular device parameter the user needs only the name of device and property. Using of them he can create a Parameter object, which serves as an access point to particular property. Afterward the get, set or

ISBN 978-3-95450-146-5

subscribe is called on the created object. The interface remains the same for all message layers.

JAPC can be easily extended with plugins to work with different middleware, which made it an easy choice for the FAIR control environment. Since the Device Access layer is also a Java based client library, GSI has developed a JAPC plug-in to add the desired functionality (see Fig. 2).

The JAPC was adopted by GSI earlier, forwarded mostly by collaboration in other areas. Hence it was tested in production and has proven to be reliable and easy to use API.

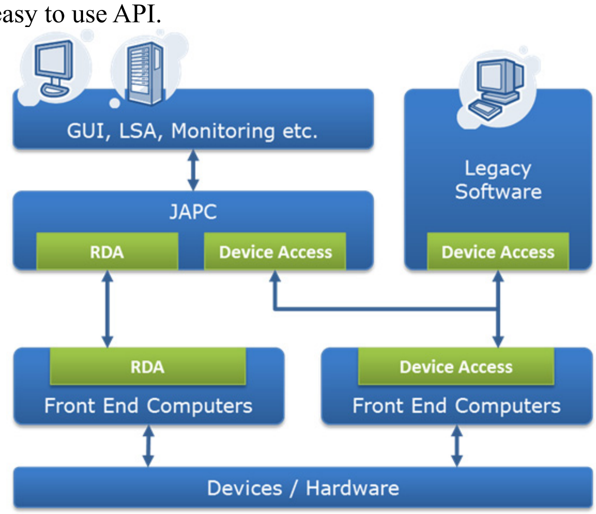

Figure 2: FAIR middleware overview.

STATUS AND FUTURE DEVELOPMENT

At the moment there are only a few devices in the running system based on the new middleware layer used mostly for the beam diagnostics. However multiple user interface applications use the JAPC API already to access particular properties. Those applications will be able to operate easily even if the accessing devices will migrate to the new middleware.

In the near feature GSI will start the operation of the CRYRING [10]. Its control system will be based on the new framework. The pre-operational tests showed that the new middleware fulfills all requirements to run this system.

The start of the FAIR operation is scheduled for 2017, hence some of the details of the future middleware system are still a topic of ongoing discussions and some components of the future system are not fully specified.

One of the future developments shall consider the access control mechanisms. Currently the Device Access relies on its own authorization and authentication mechanism, while the RDA3 uses the RBAC, which is also a part of the Control Middleware at CERN. There is no intent to run both systems simultaneously.

As for any long running project there is probably no way to forecast all the requirements, which will be discovered during the development time. However, we are sure that the RDA framework provides the necessary flexibility to fulfil those.

SUMMARY

Control system middleware is integral part of the future FAIR software landscape. With the new facility a large amount of new devices will be put in operation. The extension and further usage of the old communication layer based on CORBA technology will be a difficult task. In addition the CORBA itself can be considered as obsolete. Hence the usage of the old communication framework for the coming system would be bound with many difficulties and costs.

On the other hand the existence of a bigger amount of devices, based on older hardware, make it a costly and difficult task to migrate everything to a new base in one step.

This dilemma can be solved by using generic and middleware unaware API – JAPC, which allows the usage of multiple messaging frameworks simultaneously. Moreover it opens the possibility to change the communication layer of particular devices, without the need of adjusting of the accessing software.

The successful collaboration of GSI on FESA framework and the choice of it as the main system for development of the device software determines the usage of the RDA as the messaging layer between the control applications and devices. Its new version is based on the ZeroMQ framework and fits the requirement of FAIR middleware.

The usage of CERN components does have a great impact on the architecture of the system. So in many parts it will be derived from the existing architecture of CERN.

As it already showed by CERN, the RDA based middleware is providing the required flexibility and scalability to operate accelerator systems of a bigger scale. With it the FAIR control systems will be prepared for the coming challenges.

REFERENCES

[1] FAIR Project, website: http://www.fair-center.de
[2] CORBA, Common Object Request Broker Architecture, OMG, http://www.omg.org
[3] iMatix ZeroMQ, website: http://www.zeromq.org
[4] A. Dworak et al, "Middleware Trends and Market Leaders 2011", ICALEPS2011, Grenoble, France.
[5] S. Matthies et al, "Access Control in the Renovated GSI Control System: A Combined Name- and Rights-Server", PCaPAC08, Ljubljana, Slovenia.
[6] F.Ehm, "Running a Reliable Messaging Infrastructure for CERN's Control System", ICALEPS2011, Grenoble, France
[7] I.Yastrebov, "Role-Based Access Control for the Large Hadron Collider at CERN", International Journal of Computers Communications & Control, ISSN 1841-9836, 5(3):398-409, 2010
[8] OmniORB, website: http://omniorb.sourceforge.net
[9] JacORB, website: http://www.jacorb.org
[10] CRYRING@ESR, website: https://www.gsi.de/index.php?id=4570&L=1'