ACCESS CONTROL IN THE RENOVATED GSI CONTROL SYSTEM: A COMBINED NAME- AND RIGHTS-SERVER

S. Matthies, U. Krause, K. Höppner GSI, Darmstadt, Germany.

Abstract

The GSI control system implements an access control mechanism: Modifying of device settings is possible only when access rights are granted to the user. Originally integrated in the client side of the proprietary network layers, replacement of this middleware by the newly developed CORBA based communication required a new access control implementation [1]. Authorization is coded in device-specific patterns: A command is executed on the front-end server only when the correct pattern is provided. These patterns are handled by a central rights server, which is combined with the anyway needed CORBA naming resolution service. Being a lightweight approach, it should provide sufficient protection against undisciplined users which otherwise may severely disturb facility's operation.

INTRODUCTION

Scientific communities traditionally keep their system open. Information is freely available, and new scientific ideas should not be hampered by restrictions in equipment tuning.

However, accelerator facilities are quite complex and consequences of equipment manipulations are not always easily overseen. Erroneous or accidental manipulation of equipment may disturb machine operation severely. To reduce such risk, an access control system was added to the originally open GSI control system.

Personal safety is handled by an independent system, and an interlock system prevents physical damage of machine components. A lightweight access control implementation should be sufficient, adding only little complexity and overhead.

MECHANISM

GSI Control System

GSI control system is designed as a decentralized distributed system. Front-end computers handle the equipment and provide remote access from operation level computers. On the front-end level, the equipment is modelled according to the object oriented paradigm: devices with properties that allow to read data (read), to modify settings (write) or to execute functionality without data exchange (call).

Needs

Being aware of the full context of momentary machine operation, operators should have full access to all device properties. High level tools, handling the machines on physics parameters, ensure consistent device settings.

Additionally, machine or equipment experts and support staff often need low level access to equipment parameters. Such access should be restricted to their specific domain, where they are experts.

Basic Idea

Access rights are granted to users, authenticated by their operating system's login name. Unless explicitly granted, any critical access to devices is denied.

Classification of properties according to their criticality provides flexibility. Several levels are supported: Reading values does not cause harm, so in general read properties are at the lowest level. Modifying the settings of a device is much more critical, so in general such properties are classified by a higher level. Some properties may be so critical that they request special expertise: they may be classified even higher.

Implementation

Access control is implemented in the front-end device itself. Other than an implementation on the client side it cannot be bypassed.

The front-end system needs information whether a device access is allowed or has to be refused. Instead of handling detailed access control lists, which would request a connection to a rights database, a key mechanism is used. A 32-bit pattern, coding the access right level, has to be provided as parameter in the remote access call. Access to the property is granted only when the proper pattern is sent.

While instantiating a device on the front-end device server a set of device-specific patterns according to the different right levels is generated. This set of access patterns is transferred to a central Right-Sever. This is the only network operation a device has to perform during instantiation.

To operate on a device, the client has to create a local proxy object. The access interface in the proxy instantiation requests the pattern for this device from the access rights server and keeps it. It is added in all remote access calls.

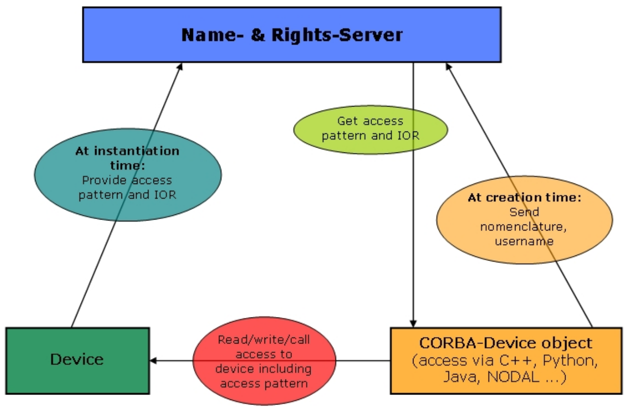

Name- & Rights-Server
At instantiation time: Provide access pattern and IOR
Get access pattern and IOR
At creation time: Send nomenclature, username
Device
Read/write/call access to device including access pattern
CORBA-Device object (access via C++, Python, Java, NODAL ...)

Figure 1: Name- & Rights-Server actions

STRUCTURE OF ACCESS RIGHTS

Device properties access rights are classified into criticality levels. According to their right level, users are allowed to call properties up to a certain criticality level only.

The access rights levels of device properties correspond to the rights levels of the Rights-Server except for the localsystem level. This level will be mapped to the system level under certain circumstances.

Criticality and Right Levels

Four different criticality levels exist for all devices' properties:

Read: Reading values will not interfere with device settings and has no influence on accelerator operation. Therefore reading is considered a harmless operation that will not change any device settings.

Modify: The property acts on accelerator operation. Its usage might affect device operations and influences accelerator operation by using modified settings.

System: Properties classified as *system* act on control system components. Therefore the operation of several devices might be affected (e.g. re-booting a computer).

Administrator: Properties classified as administrator properties are highly critical and might severely influence facility's operation. Those properties should be carefully handled by device experts only.

The Rights-Server uses the four access levels explained above. To decide about access to system properties of an equipment controller that handles devices of a given device type only the localsystem-access level is added.

Localsystem: grants access to properties of criticality system for certain control system components.

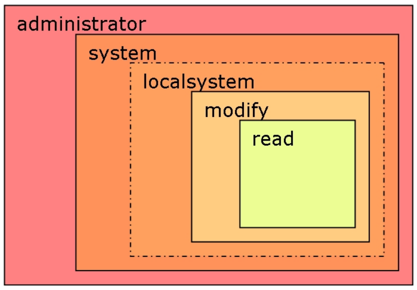


Figure 2: Grading of existing access right levels

Assigning Rights

The approvable access rights can be granted for single devices, equipment models or equipment types at specific locations. Users may have different access levels for different devices or equipment models on different locations. This access information is defined in an XML-document that is evaluated by the Rights-Server.

If no right level is defined for a specific user any access operation other than reading will be rejected.

IMPLEMENTATION

The Name- and Rights-Server retrieves its information on devices, equipment models and users from an XML-document.

Devices, equipment types and user information may be combined to groups. This kind of centralisation allows shortening the necessary definitions within the XML-document.

Even wildcards are allowed. Wildcards may be used "standalone" concerning all devices or equipment types or in combination with parts of device or equipment model names or whole areas. The valid usage of wildcards to abbreviate device names prevents extensive and potential error-prone definitions for many devices.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<root_rights
  xsi:noNamespaceSchemaLocation="http://www. ... right.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ...
    <RIGHT>
        <user group="controls">
            <admin>
                <device>DEV*</device>
                <eqmodel>EQ*</eqmodel>
            </admin>
        </user>
        <user name="Smith">
            <modify>
                <eqmodel>MX</eqmodel>
            </modify>
      <system>
        <device>DEV007</device>
        <device access="denied">DEV008</device>
      </system>
        </user>
  </RIGHT>
  ...
 </root_rights>
```

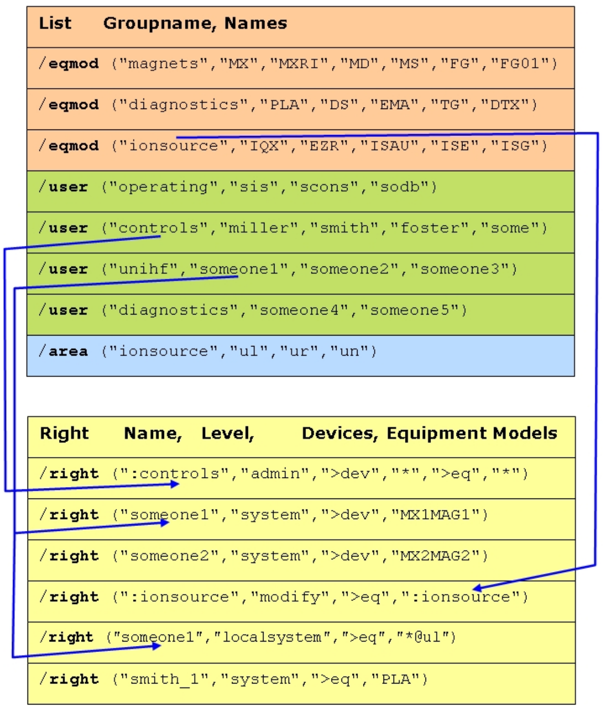
Figure 3: Extract of XML - Representation of access rights

This XML-document is interpreted while launching the Name- and Rights-Server. Its content is held as lists in memory. Different lists exist: one containing devices, another one containing equipment models; one list is reserved for users and finally another list holds the defined access right levels.

Requests to the Name- and Rights-Server are swiftly handled in separate threads by parsing those lists efficiently.

If the access rights definitions have to be updated to current requirements the XML-document simply has to be edited. Reloading the altered document can be achieved quickly by using a simple command line tool. No restart of the Name- and Rights-Server is necessary.

The following figure depicts the lists held in memory in principle. The blue arrows highlight some connections.



List	Groupname, Names
/eqmod	("magnets","MX","MXRI","MD","MS","FG","FG01")
/eqmod	("diagnostics","PLA","DS","EMA","TG","DTX")
/eqmod	("ionsource","IQX","EZR","ISAU","ISE","ISG")
/user	("operating","sis","scons","sodb")
/user	("controls","miller","smith","foster","some")
/user	("unihf","someone1","someone2","someone3")
/user	("diagnostics","someone4","someone5")
/area	("ionsource","ul","ur","un")

Right	Name, Level, Devices, Equipment Models
/right	(":controls","admin",">dev","*",">eq","*")
/right	("someone1","system",">dev","MX1MAG1")
/right	("someone2","system",">dev","MX2MAG2")
/right	(":ionsource","modify",">eq",":ionsource")
/right	("someone1","localsystem",">eq","*@ul")
/right	("smith_1","system",">eq","PLA")

Figure 4: Example of lists in memory

NAME-SERVER FUNCTIONALITY

To simplify matters the Rights-Server is combined with the CORBA-Naming Service that is used for address resolution and needed anyway.

Each device in the control system is represented as a CORBA interoperable object reference (in short: IOR).

The Name- and Rights-Server handles the CORBA IOR in its 'stringified' form. Such IOR-strings can be handled and transferred without knowledge of internal structure of the IOR. This allows accessing the devices from different platforms with different interfaces, e.g. differing operating systems or varying programming languages.

Networking

Typical TCP/IP-network operations are the base of communication of clients with the Name- and Rights-Server. The functionality used both by client and server is exchangeable, differing only in the type of packages to be sent.

Only one network operation is required while instantiating a control system device: the CORBA-IOR and the device-specific set of access patterns are delivered to the Name- and Rights-Server.

The client has to create a local proxy object to operate a device. Those proxy objects are created by a factory method. This 'device factory' requests the stringified CORBA-IOR together with the matching access pattern from the Name- and Rights-Server by sending the device name and the user name. The Name- and Rights-Server determines the user's right level for the device and codes it in the corresponding device's access pattern.

The client's local proxy object stores the requested data for all further device operations, no new request is necessary when calling another device property.

These characteristics reduce network traffic to an essential level.

CONCLUSION

Accelerator operation should not be constrained by faulty modifications of devices. A serious blockage is easily achieved accidentally when not being aware of consequences while changing important device settings.

To ensure reliable facility's operation the Name- and Rights-Server security approach limits erroneous and inadvertent manipulation of accelerator control equipment.

As any device access other than reading values has to be explicitly granted more delicate operations are limited to expert users.

The Name- and Rights-Server is implemented as a TCP/IP service with a simple protocol reducing network traffic to a minimum.

The rights management in a central service allows a flexible change of granted rights in vitro without the need for a configuration change on the front-end devices. Since checking for sufficient access rights is performed in the middleware part, no change in the equipment software on front-end computers is required.

The lightweight approach of the Name- and Rights-Server against unwanted manipulations clearly helps to ensure reliable accelerator operation.

REFERENCES

[1] U. Krause et al, "Integration of Renovated Networking Middleware into a Running Control System Environment", PCaPAC 2008