Access Rights for Device Access in the GSI Control System

Purpose of access rights in device access

The control system provides access to all devices of the accelerators. A huge number of components with a wide range of properties can be handled. Confronted with the broad spectrum of functionality, users often don't know all consequences of manipulations of the devices. It is strongly advised to limit the control system's access to components of the accelerators. A user should be able to manipulate a device only when the right to do so was grated to him.

Conflicting demands

Access rights conflict with the intentionally open system architecture. Device access is implemented by a CORBA-based interface. The interface is fully described in an IDL specification which cannot be kept hidden. Anybody who has access to the interface specification, which is the IDL specification, can implement a client application and can perform device operations.

A secure and reliable handling of access rights which cannot be manipulated would have to be implemented in the devices (the device objects) itself. This would request handling user identification, and managing of their rights, on this level. Access right information however has to be stored in a central location. Evaluation of this data in a distributed environment would lead to a rather complex implementation. The additional network traffic for the data exchange easily results in significant reductions of the performance.

Aims of access rights handling

Fortunately, a strict and watertight system to restrict, and grant, permission to access devices seems not to be needed. Integrity of human health has to be assured by other means than by general access rights management, and no precious data has to be protected. A facility used for research and education, which in addition will be constantly modified to match growing operational demands, cannot be fully protected against attacks from insiders without severely reducing its flexibility and thus its capability to service its primary focus: Providing the means for a broad range of research programs.

Therefore a lower level of security is intended:

- Users should be protected against accidentally manipulating devices.

  Unintended access may occur by simply mistyping a name. Limiting the number of components for which access is granted limits the number of errors which lead to severe consequences.

  A user may overestimate her or his capabilities and may not see all consequences of her or his actions. Granting access rights only to components, and actions, for which users is have expertise allows to assign the responsibility for all actions.

  Using the means and tools of the control system, users are permitted only to do actions for which explicit authorization was given.

- Great care has to be spent in IT security aspects. Only users who really have to deal with accelerator components should be allowed to have access to the accelerator components, including its IT elements. Keeping other people out of the system has to assured by general IT security means.
- Hardening the access rights handling against explicit hacker attacks is not envisaged. It is accepted that people with an insight in the structure of the control system, and who are allowed to act in the accelerator IT infrastructure, can circumvent the access rights mechanisms.
- However, manipulation of the access right system should not be simple. Circumventing the access rights management should request explicit activities. Especially, creating a bypass interactively by using an

interpreter or using existing control system's tools should not be practical.

Mechanism

A user based access mechanism is chosen. Rights may be granted to users, identified by their login name. Additionally, it may be necessary to grant the access only for specific computers (a user is allowed to access a device only from a specific node).

Access rights hold for a device. It will not be possible, to specify access rights on a property base (specify different access rights for each single property). However, access rights will be qualified according to property-classes. Properties then will be groups according to a security level. As a default, all read-properties form one security level (reading generally is not critical) while all write-properties form another security level (write-properties generally will change the state of a device and have to be handled with greater care than a read-access). Very critical properties, formerly handled as “ hidden properties”, may form another property group to which access then has to be granted with great care.

Access rights are stored in a central access rights data base. For each user, optionally in combination with a limitation to host computers, it contains the access rights for devices. Access to the database is by an access rights service.

Checking the user's access rights can be done on the client side and in the server side which is the device object itself. In the open architecture of the control system any client side access checking can be quite easily bypassed: Given the device's CORBA IOR, it is easy to generate an own device access implementation from the IDL definition and to interact with the device directly. Obtaining the correct IOR is nevertheless not straightforward. A query in the system's nameservice is needed which requires some nontrivial interaction. However, finding examples how to do it will not be too difficult, and proper hiding the IOR in the system parts would be difficult. IORs of the devices are used in several places in the software which makes proper screening of all parts laborious.

Evaluation of the access rights in the devices is preferred. Creating an own access interface will not help then to bypass the access checks. On the other hand, a device then needs the access right information for the specific action. Rather than requesting the access rights data base each time a device is accessed it is preferred to provide the needed information as part of the device access information, that is as an input parameter in the CORBA IDL interface. The device access information then has to be coded. It is tolerated that the access right data will not be fully protected. At least it has to be obscured sufficiently to prevent from easily reconstructing the correct information.

The mechanisms of handling access rights information is described below.

- Device access information is hold in a central access rights database. Information from the database can be queried via an access rights service.
- Data to code access rights is handled and exchanged as non-trivial bit patterns, the data patterns. A *data pattern* is as an integer of at least 32 bits. The sequence of bits in the pattern should look like a random sequence. About half of the bits should be 0, or respectivly1. Higher value bits should be used as well as lower value bits.
- Access rights like 'write access allowed', 'system access allowed', is represented as data patterns, the *access rights pattern.*
- Upon creation, each device creates a special data pattern, the *device pattern*. Device patterns don't need to be unique, but generating identical patterns for two devices should be a rare event. Device patterns are stored in a special section of the access rights database, similar to storing the CORBA IORs in the name service.
- When a user wants to interact with a device, a device proxy object has to be created. Upon creation, the device proxy queries the name service to obtain the device's IOR. Similarly, the proxy object queries the

users access rights for the specific device from the access rights service. The device access service does not return the access rights patterns directly but in a coded form as device access pattern. A *device access pattern* is the access right pattern XOR-ed with the device pattern. Identical access rights will then, for different devices, result in different device access patterns. The device proxy stores both, the CORBA IOR and the device access pattern internally for usage in device access operations.

- When the device is access, the device access pattern is send as part of the access parameters.
- The device then XOR-s the received device access pattern with its own device pattern and by this reconstructs the original access rights pattern which then will be evaluated to allow or reject the requested operation.

Critical Actions

As mentioned, the chosen mechanism is by no means watertight. Several weak points in the proposed mechanism can be seen quite easily.

- Explicit query of the access rights from the access rights service. If a user performs the required queries, with a faked user name, she or he may obtain a device access pattern to manipulate the device.
- Manipulating the device access pattern, hold in the device proxy. If a user has access to the device access pattern, replacing his access rights pattern by another one would not be difficult. She/ he simply needs the data patterns of the access right granted to her/him and the one of a more privileged access right.

To make attacks more difficult and to shift the effort to do so to a level where notable hacking energy is needed, several measures should be taken.

- Make queries to the access rights service somehow cumbersome. No interactive access to the access rights service must be foreseen for general users. As far as interactive access is indispensable, as for the system's service people, make sure by general IT means that it can be used only by people who really need.
- If possible, protect access to the access rights service such that only one special software module has access to the service.
- Don't provide user access to the device access pattern stored inside the device proxy.
- Take care that a user cannot modify the existing control system's access code. This is a special problem in interactive languages like Python.

Outline of Access Rights Structure

Access rights in the control system are granted to User names. Access may be restricted to allow access from certain host computers only.

Access rights allow access to devices. To ease handling, groups of devices can be specified in the access rights data base. This will as an example allow to specify access rights for access to

- Individual devices, identified by their nomenclatures
- Groups of devices of identical device type
- Groups of devices located in a specific area
- All devices
- All front-end computers handling a certain device type (all nodes handling profile grids or all nodes handling beam diagnostic devices)
- All resources, including all front-end computers

Access rights are layered. A higher level access right covers the access rights of all lower levels. This allows to use only one access right in each access, avoiding to handle combinations of several independent rights. Access rights may be as follows:

- Full access, including highly critical properties (which may replace the former hidden properties)
- Access to system resources
- Access to manipulate the device (write access)
- Access to read data from the device (read access)

It still has to be discussed whether additional access rights would be needed. These may comprise:

- Execute. May be needed to handle a “software device”, some software handled and accessed the same way as a piece of hardware. This may be a data base or a program, handling a special calculation.

Implementation Remarks

No detailed implementation outline will be given but some ideas will be listed how the access rights handling can be implemented

- Proper verification of the identity of the user is a very crucial part in the proposed concept. This can be achieved in several ways:
  - CORBA already provides security mechanisms. Although the CORBA security concepts are not yet understood clearly enough it seems practical to use CORBA security to receive the access rights pattern from the access rights service and verify the users identity by CORBA security mechanisms.
  - Alternatively, obtaining the user's name could by calls to the underlying operating system could be explicitly coded in the device object proxy implementation. Reliably querying the device access pattern from the access rights service then should not be implemented by a CORBA call. Explicit TCP/IP network access should be used instead. Imitating the normal query with a falsified user name then requires explicit nontrivial coding to handle the connection. Although the mechanisms to handle TCP connections are very well known the need to implement theses mechanisms explicitly shifts such attacks to a level where significant hacking energy is needed.
- Interactive languages (Python): Implement queries for the device access patterns not in the language itself but implement it in a compiled language like C and provide it as a precompiled module only.
- Since the mechanisms for obtaining the device access patterns, and to obtain the device's IOR, are quite similar both tasks should be combined. A combined device access and name server should be developed which then allows to get device access pattern and IOR efficiently in the same call.

Drawings

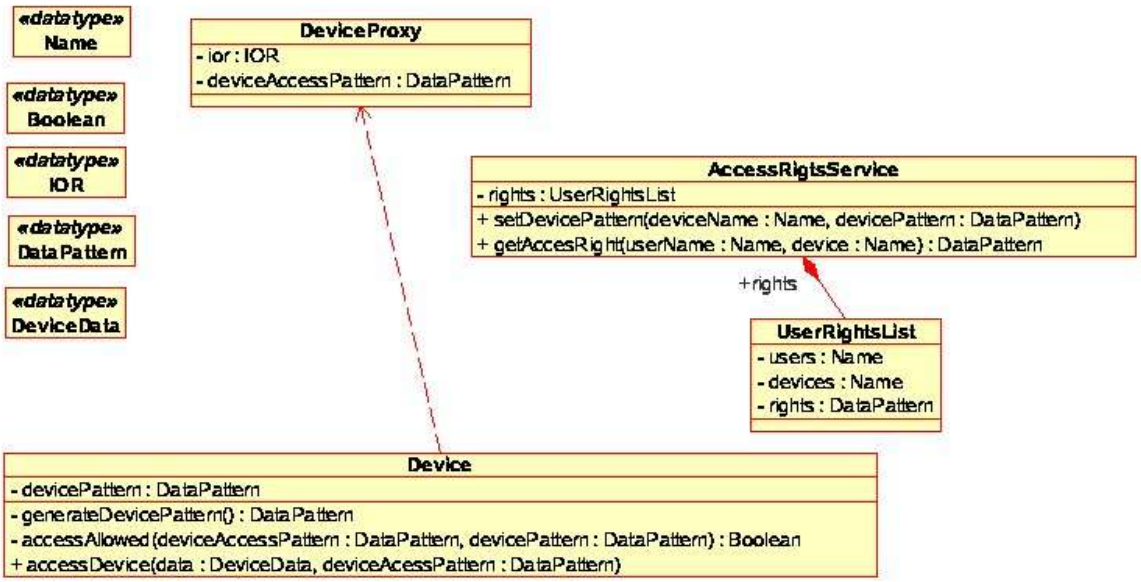

Fig. 1: Elements of Access Rights Handling

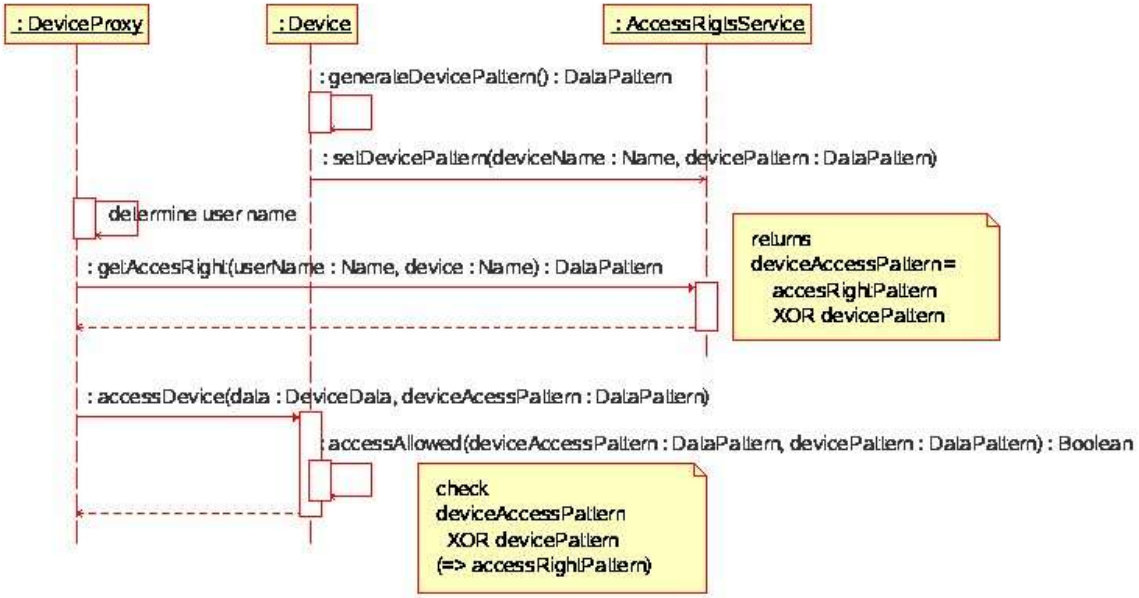

Fig.2: Activities in Access Handling