Development objects and algorithms required to implement a new method of access control to physical resources, based on qualifications

DANIELA CRISTEA[1,2], OCTAVIAN PROSTEAN[1], THOMAS MUSCHALIK[2]
OVIDIU TIRIAN[3]
[1]Department of automatic and informatics
[3]Department of Electrotechnical Engineering and Industrial Informatics
"Politehnica" University of Timisoara
5 Revolutiei Street, Timisoara
ROMANIA
[2]NWCON Technology Consulting GmbH
GERMANY
anadaniela05@yahoo.com, octavian.prostean@aut.upt.ro, t.muschalik@nw-con.eu, ovidiu.tirian@fih.upt.ro

Abstract: - This paperwork presents a distributed system to be used for implementing a new method for the access control to physical resources, along with the mode to implement various development objects required, and the algorithms used for encoding and decoding the access right. This access control method, described through the QBAC pattern, is going to be used by the employees of a company to access the inputs of any machines, based on the qualifications (abilities) they dispose of.

Key-Words: distributed system, authorization, access control, security, SAP NetWeaver platform, qualification

1 Introduction

In nowadays society, the computer system attacks are real threats, and terms as: *hacker, SPAM, phising, cyberterorism* are very often heard.

The experts in this field trigged a warning signal regarding the escalation of the cases when, through different techniques, someone goes for obtaining the authentication data of a certain entity (e.g. person, system), in order to abuse of its authority or, exploring the weaknesses of a system, to realize different types of attacks. The security lacks provoke important damages, situations about what we can often read on the first page of the world's well-known newspapers.

According to a statistic performed by CSI [1] based on the interviews with 144 organizations that agreed to offer information in this respect, the damages due to various security problems (e.g. viruses, unauthorized access to the systems) amounted, in 2008, approx. $288.618 per interviewed entity. The possible attackers of a system can be not only the hackers, criminal organizations, etc., but also the employees or former employees of a company that know very well the respective system. That's why the security should not be neglected, even in case of the functionalities created for the employees.

The protection of the information and the security of the access areas are daily necessities, this being the reason why the access control to resources plays a more and more important role. These ones, along with other methods (e.g. Cryptography, Firewall), ensure that an entity can access only those information or only those physical resources for which it holds the adequate authorization. The access control to resources (physical and informational) is realized based on authentication and authorization. But, in the same time, we should take into account the fact that the usage of a solid authorization concept or an authentication method with high security degree doesn't necessarily mean the obtaining of a secure system. All these should be combined with the usage of other adequate security methods, with a secure programming able to avoid the vulnerabilities and to keep off the eventual attacks.

Hereunder, we present a model of access control to physical resources, based on qualifications that can be obtained through a learning process. The development of this model was imperative for granting to the employees of a company the possibility to access the inputs of certain machines (protected objects), according to the qualifications (abilities) they dispose of. To meet the requirements of our project, we have studied the design patterns existent in the field of patterns used for the access control to resources, but we found that none of the existent models fulfils completely the requirement

of the project. That's why it was necessary to develop our own solution: the access to the inputs of certain protected objects to be realized based on qualifications – QBAC [2]. The solution that grounded the access control method used in this paper has been inspired from the patterns [3]: Session, Extended Authorization, RBAC (Role Based Access Control), MBAC (Metadata Based Access Control) and Access Control to Physical Structures [4]. Besides a combination of their basic ideas, it was necessary to add our own elements, as the access based on qualifications or the addition of certain security elements for the physical level (e.g. to lock/unlock data about machines and employees).

The purpose of the present paper is to present the structure of the distributed system required to implement its component elements in this model, along with the implementation modality.

2 The structure of the distributed system

Nowadays, the Internet plays a special role, offering to the employees either the access to secured networks (from distance) or the possibility to control, by Internet, certain industrial processes. The closed and simple systems, where the access to resources is locally realized, are not a common scenario anymore. One of the necessities brought by the globalization is the access to the resources of a corporation from all its centres spread in different parts of the world, the necessity to realize distributed industrial systems, the usage of standard protocols and more and more complex networks. This led to the extension of the industrial systems where, to ensure the security, the IT team should cooperate with the system engineers.

The structure of the distributed system for implementing the method of access control based on qualifications is presented in Fig. 1.

At the controllers' level, we used PLC (Programmable Logic Controller) of Siemens family. At this level, it is possible to connect those "n" machines whose inputs can be accessed by the employees, based on the qualifications they dispose of. Each machine has its own RFID card reader that offers to the employees the possibility the login and logout. For programming and configure the PLC, we used "Step 7" software that offers some facilities, e.g. testing, diagnosis, on-line function to display all the used variables. "Step 7" disposes of three basic programming languages, to which it is possible to add more languages, to extend its functionality.

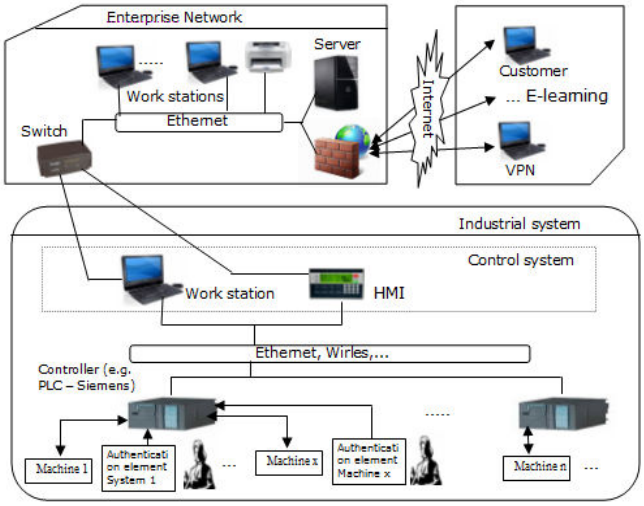

Fig.1 Structure of the distributed system for implementing the access control model based on qualifications

In the present project, we mostly used the S7-SCL language (useful in case of complex algorithms), along with the FBD language. So, we created a modular structure by using various types of blocks and combining various types of programming methods, according to the requirements.

All the development objects needed for the access control method based on qualifications are created at the server level. For the server part, we chose the SAP NetWeaver platform [5], because it offers certain advantages, as follows:

- Large variety of so-called "usages" to apply;
- It is the base of SAP Business Suite that includes the SAP ERP HCM (Human Capital Management). By using this, we can create the entire process afferent to human resources and organizational structure of the company, along with the qualifications, courses and the whole learning process;
- Application Server ABAP and Application Server Java;
- Portal through SAP NetWeaver Portal;
- MVC (Model View Controller) support;
- Easiness to work with Web Services and ActiveX;
- We dispose of various tools for creating the Multilanguage applications, to make the created applications to "speak" the languages of the users they address to;
- We dispose of a large range of tools and techniques that support us in the programming process.

At the Work Station level, in the area "Control system", it is realized the communication between PLC and server, having the possibility to access certain process values. In our project, we used a Web Service for the communication with the SAP NetWeaver platform. So, in a Visual Basic application, it is consumed the Web Service that are going to grant access right to an employee to the inputs of one of the protected objects. The access right obtained from the server will be transmitted encoded in an integer, from where it will be decoded at the PLC level. In the same Visual Basic application, it is used an ActiveX (client's OPC) meant to communicate with an OPC server [6]. All these are schematically presented in Fig. 2.

So, by using Web Service, we will obtain data from the server anytime when an employee wants to login to one of the machines, and we will insert some values at the server level, e.g. the activity of a subject during the login period, date and time of logout, the machine where he was logged, etc. By using Web Service, we can also offer these data in case that certain machine is distributed outside the Ethernet.

Another possibility to communicate between the server and PLC, through the control Work Station, is to use the ActiveX (client's OPC) directly in the ABAP coding, communicating in this way with the OPC server, to read and write data in PLC.

The other Work Stations located in the area "Enterprise network" are used to create all the development objects required, at the server level, by the administrators of this model, the employees that work in the company's offices, etc.

The company's employees and clients can access the server from the outside, through Internet. For example, a client can read about the company's offers, while an employee of the company can program something in the integration platform, by using the VPN connection.

In Fig. 3, we present the main operations required for the authorization concept based on qualifications and physical connection through the RFID, which is the basis of this project and whose implementation we are going to present hereinafter.

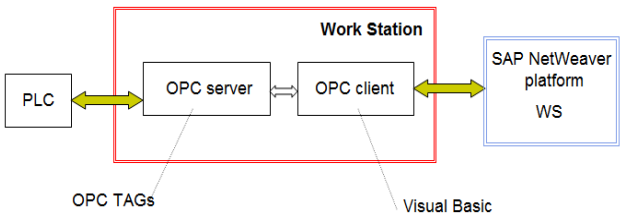

Fig. 2 Schematic representation of the communication PLC – server, through Control Work Station

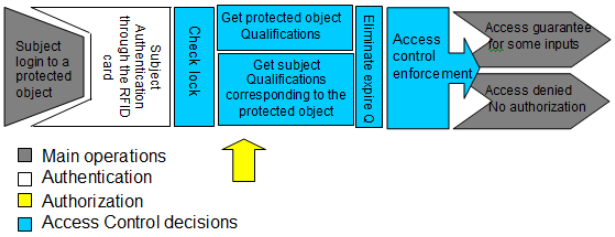

Fig. 3 Access control method

3 Creating development objects at the server level

SAP NetWeaver is the SAP integration and application platform that offers modern solutions to the companies to control the economic processes. It is made of four levels: application platform, process integration, information integration and people integration. The application platform plays a central role. Without it, we don't have programming environments, wizards, frameworks, etc., and the other components (ex. SAP NetWeaver Portal, SAP ERP HCM) can't work. The application layer is made of the Application Server (AS) ABAP and AS Java. To create ABAP applications, we have ABAP Workbench, and for creating Java applications we have SAP NetWeaver Developer Studio.

So, after choosing the SAP NetWeaver platform, we had to choose the application server (ABAP or/and Java) to be used to implement the development objects required for the scope of this project. Because we needed the SAP ERP HCM component, it was necessary to choose the ABAP application server, taking into account the fact that this component is using AS ABAP, for storing data. It was possible to apply AS java to the rest of the logic (by using a connection between the AS ABAP and AS java), but this could have a negative influence on the execution speed. This was the reason why we have chosen the ABAP as application server, creating here all the development objects required to implement the access control method.

We are going to use also the Java application server, but in a transparent and involuntary manner, without being necessary java programming. For example:

- The SAP NetWeaver Portal component runs on the Java stack. The possibilities to realize the communication with the ABAP back-end is automatically offered by this component.
- When we will test a Web Service in the framework of the ABAP Workbench, we

use a test page (Web Services Navigator) that runs on the Java stack.

- When we need ADS (to create Adobe forms in Web Dynpro ABAP), we'll still use AS Java, because this one is installed on the Java stack.

In Fig. 4, there are presented the modules and parts of SAP NetWeaver used for the scope of this project. So, by using the SAP NetWeaver portal, we are going to realize the extended variant QBAC, and to integrate the interface level management application with the machines. By using the SAP ERP HCM mode, we can create the subject related processes (e.g. working place, personal data, qualification and re-qualification, courses), and by using the AS ABAP we are going to create all the development objects required for the access control method.

The entire process started with the creation of the organization structure of the company, followed by the creation of employees' data and the data required for the learning process (e.g. courses, qualifications), realized by using the SAP HCM module. This component is one of the most complex ERP components; it disposes of a large range of functions (components). In our case, we used:

- Organization Management OM that offers the possibility to create the company's data;
- Human Resources HR that offers the possibility to create the company's subjects;
- Learning Solutions LSO, Personal Development that offers the possibility to create the qualifications and the entire afferent process.

By using the SAP ERP HCM component, we are not only able to create the required data without being necessary any programming effort or creation of the corresponding databases, but we also benefit of some advantages, as follows:

- An adequate authorization concept able to protect the created data;
- Advanced search capabilities: Reports and Queries: InfoSet, Ad Hoc Queries;
- Support for the particularities of each country;
- Possibility to be adapted to the requirement of each company, through customizing (standard) or enhancement (as specific requirements for a particular client, which are not part of the standardized ones).

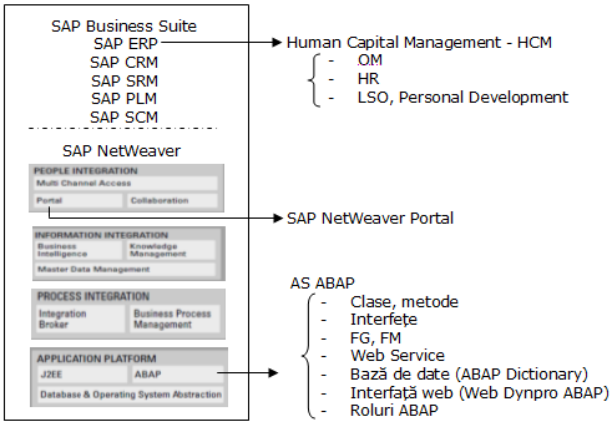

Fig. 4 Used modules and tools

Of a special importance are the data key of each employee (an identification number made of 8 digits) and the key of each qualification. The identity of an employee in the company is represented by this unique key that is going to be also imprinted on the identification RFID [7] cards. For each machine, we created three qualifications: Installer, Operator and Tool_Setter. Then, to each qualification we will assign a number of inputs of the respective machine.

So, when the employee disposes of a qualification, he/she has the authorization to serve the respective inputs of the protected object. For each machine, an employee may dispose of multiple qualifications and a qualification has the property that can expire. In Fig. 5, we presented the Qualification catalogue of four test machines. The relation *subject - qualification - access right* is presented in Fig. 6.

To obtain the qualifications of each machine, we have created a catalogue of courses.

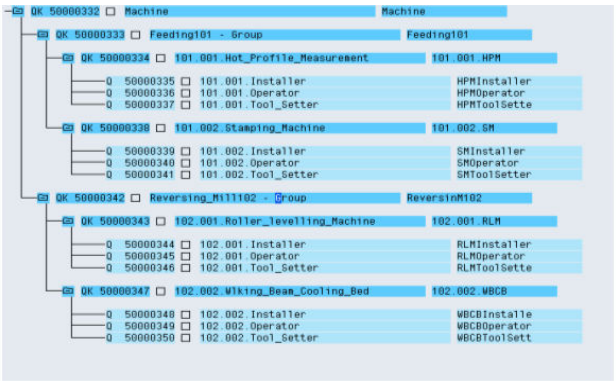

Fig. 5 Qualification Catalogue

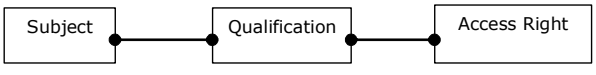

Fig. 6 Relation Subject - Qualification - Access Right

For each course, we can opt for a certain type of schooling: classroom training, web-based training, virtual learning room, online test. In the same time, each course disposes of some proprieties that can be set, e.g. the calendar schedule of the course, the cost of the respective course (for the internal and external employees), its location, language, minimum/maximum/optimum capacity. In Fig. 7, it is presented the courses catalogue for the test machines in the QBAC variant, where the schooling is of *classroom training* type. For the variant QBAC extended with E-learning [8], the courses are of *web-based training* type.

In this project, the QBAC pattern has been combined with the RBAC pattern, to offer modern learning possibilities to the employees and the possibility to make reservations for courses through the company portal and other services of *ESS* (Employee Self-Service) type (Fig. 8).

To realize the functionalities required for the QBAC logic, we created development objects, as follows:

- Relational database;
- Classes and methods;
- Access right exposed as Web Service;
- Web Dynpro application to manage the machine integration;

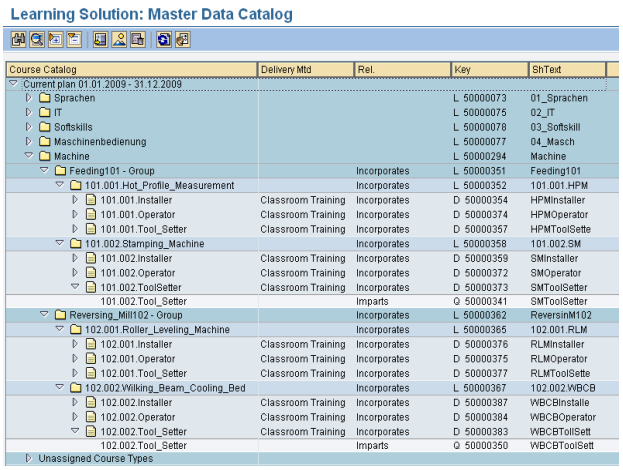

Fig. 7 Courses catalogue – The classroom training variant

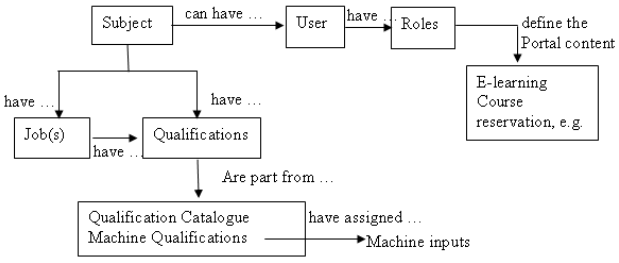

Fig. 8 Relation QBAC - RBAC

- Portal object required to make reservations to a course, to integrate the Web Dynpro application or to perform E-learning.

In this paper, we are going to present only some of the development objects we have created.

For storing the data required for the access control method, we used relational databases, and to work with them we used SQL. Because the DDL (Data Definition Language) declaration of *create*, *alter* or *drop* type is not supported, we used the tools offered by the ABAP Dictionary. The scope of the database created is to realize the connection between the machines (their inputs) and the database generated when creating data in the SAP HCM module. In Fig. 9, we present the part that integrates the machines in the access control process, the table PA0002 and the viewZ_HRP1000, making the connection with the HCM data.

Besides the tables, it was necessary to create global data elements, views, domains, search helps and table types. The Z_HRP1000 view has a special importance, because through it we created the link with the qualifications. We opted to use a view instead of the HRP1000 table, because we realized a selection according to the Q (Qualifications) type, ignoring the other objects of the system (e.g. objects: D, L, C, etc.).

By using the ABAP language [9], we created the logic required to encode the access control method. This is the SAP owner language that offers certain advantages, as follows:

- It procedurally supports OOP;
- There are integrated functionalities, e.g. SQL, SAP LUW, RFC, XML;
- It offers Multilanguage support;

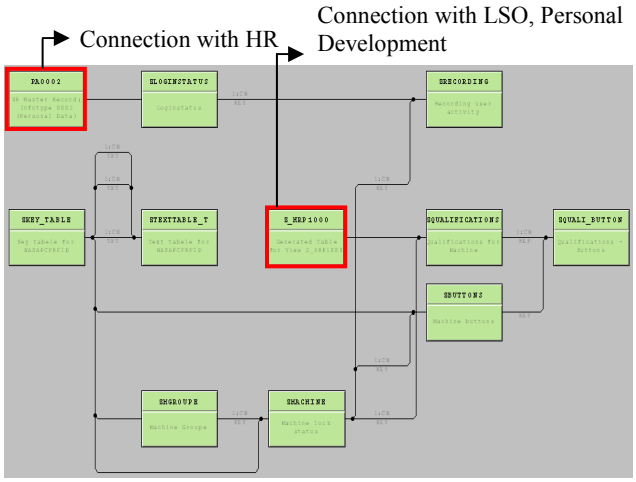

Fig. 9 The database required to integrate the machines in the access control method

- Large variety of declarations that can be used (e.g. field-symbols, call transaction, authority-check);
- The created development objects can be tested in their own environment;
- Large variety of testing and code analysing tools (e.g. ABAP Debugger, Syntax check, code inspector, ABAP Runtime Analyses).

To realise the secure applications, we guarantee that we have done the following basic operations:

- Filtering the user's inputs, even if these users are the employees of the respective company, removing the SQL injection and other attacks that can occur in this way;
- Verifying the users' authorisation, including the one connected through the Web Service, removing the eventual backdoors;
- By using the Web Dynpro ABAP technology, we reduce the number of the eventual attacks that can act on a UI technology, according to [10].

The class structure used to implement the logic of the access method is identical with the one presented in the QBAC pattern, respective a class for the login functionality, these classes inheriting a global class that represents, in the same time, an assistance class (Fig. 10).

For the scope of this project, we used either the OOP programming or the procedural one, combining the two methods for well-grounded reasons. To realise the QBAC logic, we created classes and methods by using the ABAP objects, and for offering the login/logout sessions through a web service it was necessary to create a Function Module (by using the classical programming). This combination was necessary because a Web Service (of *inside-out* type) can be created by using the ABAP Workbench only from a Function Module, Function Group, BAPI or Interface Message.

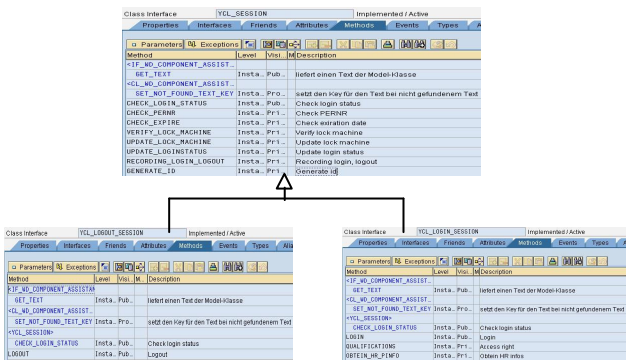

Fig. 10 The class structure used for implementing the logic of the access control method

Definitely, we could use the SAP XI component to avoid the classical programming. But, taking into account the fact that we need a single Web Service, the usage of a new SAP component for a single functionality is not justified.

A Function Module is part of the classical programming and can be independently created only in a Function Group. All these represent ABAP programs of a certain type, which are very much used for programming the SAP GUI (Graphical User Interface). By using the created Function Module, we are going to offer the possibility to login/logout in the system, by calling the public methods that correspond to the above-mentioned created classes. Indifferently if the login/logout is realised through a Web Service or directly from the ABAP (through ActiveX communication with the PLC), this Function Module is used as login/logout basis to/from the system. In Fig. 11, we present the codification section of the Function Module we have created.

With Web Dynpro ABAP, we created the web application used to manage the level of the interface with the machines. The Web Dynpro ABAP is part of the AS ABAP presentation section. Besides the Web Dynpro ABAP, in the AS ABAP presentation zone we have BSP *(*Business Server Pages) and SAP GUI. We can't create web application by using the SAP GUI; this is the SAP technology developed to offer the classical graphic interfaces, also named "Dynpros". With BSP, we can create web applications, because this is the technology created before the Web Dynpro ABAP, which combines ABAP and HTML. But, this technology has many disadvantages and a complex structure. That's why we decided to use Web Dynpro ABAP, the new SAP technology that helps us creating web applications based on the MVC principle. Some of its advantages are [11]:

- WYSIWYG view editor;
- Componentisation of the applications we have realised;

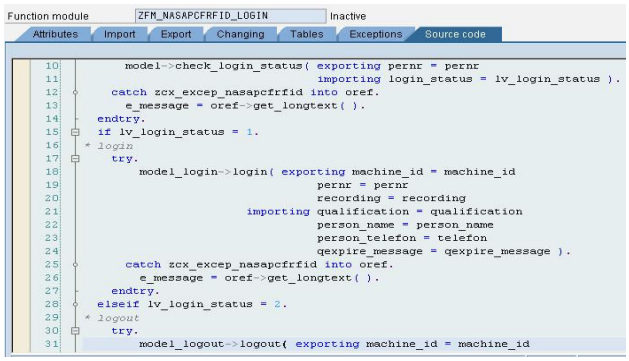

Fig. 11 Function Module, as the basis for login/logout to/from the system

- Various standard methods, the so-called "Hook methods", that can be used to interfere in certain moments during the program execution;
- Standard components (e.g. SO - WDR_SELECT_OPTIONS, ALV - SALV_WD_TABLE) that can be used to ease the programming work;
- Static and dynamic programming;
- Large variety of UI elements put at our disposal;
- It doesn't require any knowledge about HTML or JavaScript; it is sufficient to know the ABAP language and how to work with the Web Dynpro Framework. Then, during running, the complex algorithms will transform the application into HTML, JavaScript or XML code;
- Possibility to use the new technologies. (e.g. Adobe Forms);
- Possibility to personalize the applications we have realised.

The modality to implement the administration Web Dynpro application is presented in Fig. 12. From the basic functionalities, we mention only the possibility to create different reports in PDF format (to track the employees' activity) and the possibility to import the names and inputs of the machines from text files.

As we have mentioned above, the functionalities of login/logout to/from the system are also offered through a Web Service. Nowadays, we use more and more the Web Services, from E-commerce functionalities to automation. With ABAP Workbench, we can be either a web service provider or a web service consumer.

In our case, we are a web service provider that uses the created Function Module as an endpoint.

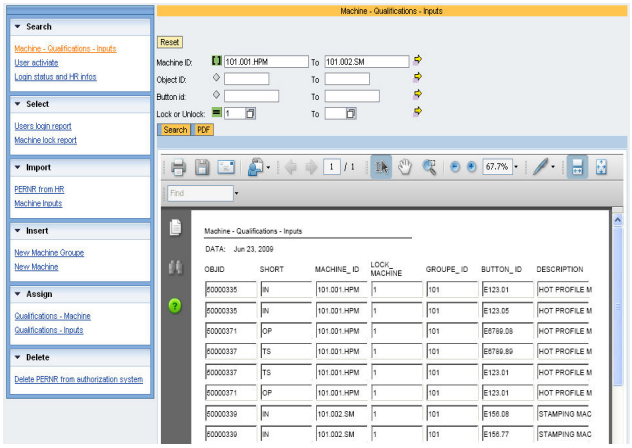

Fig. 12 Web Dynpro management application

Some of the advantages to use a Web Service for the present projects are:

- It is available through Internet or Intranet (Ethernet), being easily consumed in the Visual Basic applications (or another application type), from the control level;
- It is independent of any programming language, because it uses the XML grammar; it can be also used at the PLC level.

The created Web Service is provided with:

- **inputs**: machine ID (the ID of the machine to which the login/logout is realised), PERNR (the ID of the subject that wishes to login/logout to/from the respective machine), and recording (the eventual activity of a subject to one of the machines – for the login case);
- **outputs**: PersonName (the name of the subject who logs-in), AccessRight (the access right codified in an integer), telephone (the telephone number of the subject who logs-in), Smessage (success message in case of a successful logout), Emessage (error messages), QExpireMessage (messages in case one or more qualifications are going to expire) and ExpireDate (expiry date of one or more qualifications).

The access right transmitted from the server to the PLC shall be codified in an integer, to avoid the overloading of the communication network (at the communication through web service) and to define a lot of global values in PLC (in case of using the ActiveX directly in the ABAP language). In the next subchapter, we present the coding and decoding algorithm specially realised for this purpose.

With the SAP NetWeaver platform, we have the possibility to create Multilanguage applications able to offer the required support for the desired languages, without being necessary to recode them. In this respect, SAP put at our disposal a range of tools. When using them, we have to avoid introducing in the code the specific strings of the texts to be offered in more languages. In the same time, in case of AS ABAP, the languages used to offer support should be installed in the system. All the development objects realised in this project are Multilanguage, with the initial language English and support for German. It is very easy to offer support for other languages, because all we have to do is to translate the centralised strings.

Because we need to register in the database the description of the machines, the machine groups and

the buttons of each machine, in order to offer them to the user in the desired language, it was necessary to create two extra tables, ZKEYTABLE and ZTEXTTABLE_T, to be used for storing the respective strings according to the language. The strings firstly inserted in English can be translated later in other languages (German, in our case). Then, the selection of the language will be realised according to the login language used by the person who wants to see the data, realising actually a selection according to the SPRAS language field.

To realise the messages and the Multilanguage exceptions, we used message classes and exceptions, as well as strings inserted in the assistance classes. By using the tools offered by the ABAP Workbench, we translated the respective strings, and the messages are going to be displayed according to the login language. In the same time, the qualification and courses catalogues have been created as "Multilanguage", by using standard tools put at our disposal by the environment.

To realise the Web Dynpro application as "Multilanguage":

- we used the OTR (Online Text Repository);
- we translated the description texts of the global data objects, fields and other development objects realised in the ABAP Dictionary;
- we used texts included in the model assistance class;
- we used the exception classes and messages;
- we didn't use static strings to define the properties of the applied UI elements;
- we translated the texts from the Adobe forms.

4 Work sessions at the PLC level

To send the commands to the machines and to receive information from them, we need certain algorithms and steps to follow, for realising operations as: reading the identification number from the subject, determining the ID of the machine where the user wants to login, deciding between logging-in and logging-out from the respective machine, sending data to the server, decoding the employee's rights for obtaining the physical addresses where he/she has access right, making commands, registering the employee's activity at the machine where he/she has logged-in, etc.

For all these, we dispose of three work sessions at the PLC level (Fig. 13):

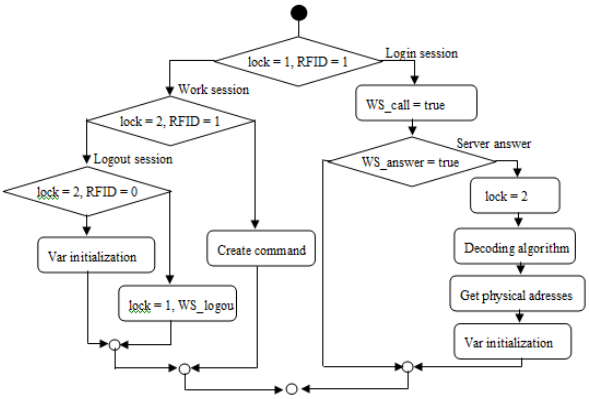

Fig. 13 PLC sessions

- **Logging-in session**, that reaches the server. Through this session, we are going to establish the right of a subject at the respective protected object. After decoding the right of the subject and re-obtaining the physical addresses of the inputs to which the subject has access right, the next step is the working session;
- **Working session**. This session is local, only at the PLC level. In this session, the subject can access only to those inputs for which he/she has the required authorisation;
- **Logging-out session,** that reaches the server. The operations performed in this session are: unlocking the subject, unlocking the protected object, registering the subject's activity in the database.

5 Coding and decoding algorithm

The coding in an integer of the inputs to which a subject has access right is very useful, because we don't have to transmit the physical addresses of the respective inputs between the server and the PLC. A protected object may have a very large number of inputs: when this number increases, the transmission of the inputs becomes more and more difficult.

For the codification of the employees' access rights, we used the hierarchy of the physical addresses of the machine inputs and the binary-integer transformation. For example, if the i protected object disposes of j inputs: $I_1....I_j$, we will consider that I_1 represents the less significant bit, and I_j represents the most significant bit. So, the formula to calculate the access right (right - R), after removing the redundancies and the expired qualifications, will be (1).

$$R = I_1 2^0 + I_2 2^1 + I_3 2^2 + ... + I_j 2^{j-1} = \sum_{i=1}^{j} I_i 2^{i-1} \quad (1)$$

The bits that correspond to the inputs for which a subject has access right will be 1, and the rest of the bits will be zero. Therefore, during a login session, we obtain the subject's access right codified in an integer.

At the PLC level, for re-obtaining the physical addresses, we have to firstly decode this right (Fig. 14). So, we will re-obtain the adequate bit series in the vector "a[j]" (1 = access right; 0 – no access right).

After decoding, we can determine the physical addresses of the inputs where a subject has access right and the addresses of the inputs that are going to be commanded in case the employee presses one of the allowed inputs.

For example, for a PLC of Siemens family, the inputs and outputs are divided in groups of 8 digital inputs or outputs:

- Ix.y, where "I" represents the input address type, "x" represents the address byte, and "y" represents the address bit;
- Qz.k, where "Q" represents the output address type, "z" represents the address byte, and "k" represents the address bit.

We can take a simple example: for a machine "x" we have stored in a vector the addresses of the inputs it disposes of (2):

$$Imachinex[10] = \{I4.0, I0.1, I6.2, I0.3, I0.4, I0.5, I0.6, I0.7, I1.0, I1.1\} \quad (2)$$

For the same machine "x", we have the addresses of the machines stored in a vector. These addresses are commanded when the respective input is active (3):

$$Qmachinex[10] = \{Q1.0, Q2.1, Q3.2, Q1.3, Q1.4, Q1.5, Q1.6, Q1.7, Q2.0, Q2.1\} \quad (3)$$

If the access right of the "xx" subject received from the server is R = 14, after its decoding we will obtain the vector (4).

$$a[4] = \{0, 1, 1, 1\}. \text{ In this case, } n = 4 \quad (4)$$

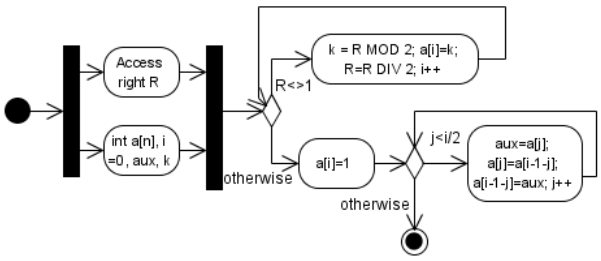

Fig. 14 The decoding algorithm

We are going to use a mini algorithm, to obtain the physical addresses of the inputs where the employee has access right and the physical addresses of the afferent outputs that are going to be commanded (Fig. 15).

So, in the vector "AddressI[j]" we will obtain the addresses of the inputs to which the employee has access right (which correspond to his/her qualifications).

In the vector AddressQ[j], we will obtain the addresses of the outputs that can be commanded when the afferent input is 1, i.e. a button is pressed. In our case, for the machine x, with the access right R = 14, we will obtain the following values (5):

$$AddressI[3] = \{ I0.1, I6.2, I0.3\}$$
$$AddressQ[3] = \{Q2.1, Q3.2, Q1.3 \} \quad (5)$$

Therefore, when:

I0.1 = 1 -> Q2.1

I6.2 = 1 -> Q3.2

I0.3 = 1 -> Q1.3, the rest of the protected object inputs are without authorisation for the subject "xx".

In this way, we will be able to decide who has access right and which is the subject's right. At the PLC level, we will not know which is the employee's right, these rights being stored, as we have seen before, at the server level. But, at the PLC level, we should have the same ordering of the machine inputs as in the server part; otherwise, the entire logic will not work properly.

6 Further research

The access control model presented in this paper has been implemented only one time, as prototype; further developments are going to be especially realised at the hardware level. The server level algorithms and the PLC are generalized created, for the "n" machine, but for testing at the hardware level we used only one test machine and one PLC Simatic S7-300, with the modules: RFID interface, Inputs, Outputs, Ethernet.

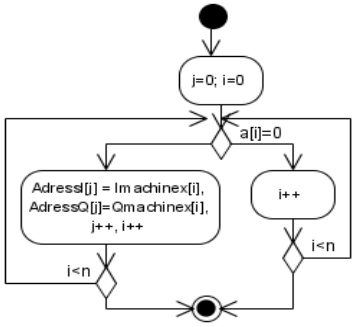

Fig. 15 Obtaining the physical addresses

In Fig. 16, it is presented the hardware structure used to test this access method. Because, in case of the test machine, the inputs are represented as sensors, we created a small control panel. By using this, we will be able to realise the order only after determining the right held by a subject towards the respective protected object.

Further developments shall be realized on the security part, where certain improvements are required, e.g. to secure the Web Service with digital signatures.

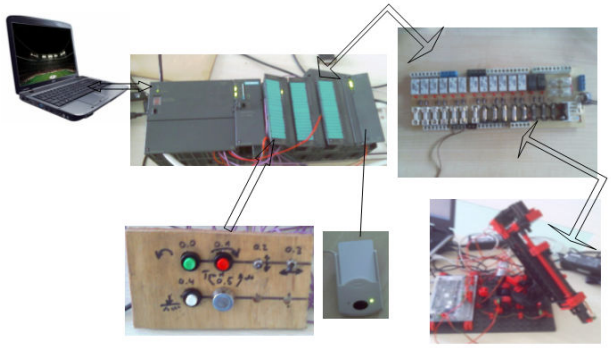

Fig. 16 Test hardware structure

7 Conclusion

This paperwork describes the implementation method of the access control to physical resources, based on qualifications. There are more advantages of this model than the access to resources based on qualifications, as follows:

- The integration of the qualifications in the authorization process helps us in the human resources process, too. So, we can easily answer to questions of this type: What qualifications are missing to an employee to perform the operator job at the machine x?
- Combining QBAC with RBAC, the employees are offered modern learning modalities
- Sustain directly the continuous learning process of the employees

In conclusion, if we look at the obtained results from the point of view of a distributed system, we can say that we obtained a system: transparent (the components are well combined, making a whole), open (it uses protocols and standards, flexible and easy to configure or to add new components), quasi secure (future developments will cover the lacks of the test variant). Regarding the Scalability and Concurrence, they are going to be tested in the next variant, when we will benefit of sufficient number of test machines.

References:

[1] Robert Richardson, CSI Director, 2008 - *CSI Computer Crime & Security Survey*, available online: http://www.cse.msstate.edu

[2] Cristea Ana Daniela, Prostean Octavian, Muschalik Thomas and Tirian Ovidiu, *An access control pattern based on qualifications to grant access to physic resources*, The 3rd European DAAAM International Young Researchers´ and Scientists´ Conference 25-28th, Vienna, Austria 2009, ISBN 978-3-901509-70-4, p. 1765 – 1766.

[3] Schumacher M., Fernandez-Buglioni E., Duane Hybertson, Buschmann F. and Sommerland P., *Security Patterns Integrating Security and Systems Engineering*, John Wiley & Sons, Ltd, 2006, ISBN: 0-470-85884-2.

[4] Fernandez, E.; Ballesteros, J.; Desouza-Doucet, A. & Larrondo-Petrie, D. (2007). *Security Patterns for Physical Access Control Systems*, in: Data and applications security XXI, Barker, k. & Ahn, G. (Eds), 259-274, Springer, ISBN: 978-3540735335, Germany.

[5] Martin Raepple, *The Developer's Guide to SAP NetWeaver Security*, SAP Press, ISBN 978-1-59229-180-9.

[6] Schwarz M. H. and Boercsoek J., *OPC for Process Maintenance*, 6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, Tenerife, Spain, December 14-16, 2007 (p. 237-243).

[7] S. Srinivasan, Aggarwal A. & Kumar A., *RFID Security and Privacy Concerns*, Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005.

[8] Egil Ginters and Arnis Cirulis, *Virtual environment use in e-learning*, 6th WSEAS International Conference on E-ACTIVITIES, Tenerife, Spain, December 14-16, 2007

[9] Razvan Bologa and Ana Ramona Lupu, *Accelerating the Sharing of Knowledge in Order To Speed Up the Process of Enlarging Software Development Teams - a Practical Example*, Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19, 2007

[10] Wiegenstein A., Schumager M., Schinzel S. and Weidemann F., *Sichere ABAP Programmierung*, SAP Press 2009, ISBN 978-3-8362-1357-8.

[11] Gellert Ulrich and Cristea Ana Daniela, *Web Dynpro ABAP for Practitioners*, Springer, 2010, ISBN 978-3-642-11386-4.