CONTRIBUȚII LA CREAREA ȘI DEZVOLTAREA UNUI NOU CONCEPT DE AUTORIZARE SAP BAZAT PE CALIFICĂRI ȘI CONECTARE FIZICĂ PRIN INTERMEDIUL RFID

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul INGINERIA SISTEMELOR
de către

Ing. Cristea Ana Daniela

Conducător ştiinţific:	prof.univ.dr.ing Octavian Proștean
Referenţi ştiinţifici:	prof.univ.dr.ing. Paul-Șerban Agachi
	prof.univ.dr.ing. Mihail Abrudean
	prof.univ.dr.ing. Mircea Vlăduțiu

Ziua susţinerii tezei: 12.07.2010

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor
12. Ingineria Sistemelor

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

După cum și alți doctoranzi au scos în evidență, rezultatele unei activități doctorale sunt publicate în diverse jurnale sau au fost prezentate la diverse conferințe naționale și internaționale. Astfel, teza de doctorat reprezintă un mănunchi de articole ce scot în evidență contribuțiile aduse într-un anumit domeniu, articole la care se adaugă totodată și alte elemente necesare descrierii în întregime a problematicii studiate.

Principala contribuție adusă de autor este metoda de acces controlat, metodă ce a fost sintetizată sub forma unui nou pattern numit QBAC. Acest pattern este recunoscut international fiind mentionat și de organizația pattern-urilor pentru securitate. Prin intermediul pattern-ului QBAC se oferă o soluție a unei probleme ce îmbină două teme actuale și anume: accesul controlat la resurse fizice și necesitatea realizării unui sistem ce determină și totodată oferă angajaților posibilitatea de participare la un proces continuu de învățare.

Regretul meu este că pe parcursul studiului doctoral nu am găsit colaboratori români care să aibă preocupări în domeniul accesului controlat la resurse fizice sau/și informaționale cu implementare directă SAP.

Înainte de a încheia, aș dori sa aduc mulţumiri deosebite tuturor celor care m-au susținut pe parcursul „drumului doctoral”. În mod special doresc să îi mulțumesc însă conducătorului de doctorat, prof.dr.ing. OCTAVIAN PROȘTEAN, precum și unui profesor cu deosebite capabilități pedagogice, prof. dr. ing. MIRCEA VLĂDUȚIU. Aduc de asemenea mulțumiri domnilor prof. dr. ing. PAUL-ȘERBAN AGACHI (Universitatea Babes-Bolyai, Cluj-Napoca) și prof. dr. ing. MIHAIL ABRUDEAN (Universitatea Tehnică din Cluj-Napoca), care au răspuns solicitării de a face parte din comisia de analiză a tezei, pentru răbdarea de a evalua prezenta teză.

Totodată, vreau să mulțumesc firmei NWCON Technology Consulting (Germania) și în special lui THOMAS MUSCHALIK care mi-a pus la dispoziție toate uneltele (sistem SAP, hardware) necesare implementării și testării metodei de acces dezvoltate. Nu în ultimul rând, vreau sa îi mulțumesc lui ULRICH GELLERT (S+P Lion Germania) alături de care am publicat în editura Springer două cărți având ca și subiect una dintre tehnologiile SAP folosite pentru implementarea metodei de acces creată.

Timişoara, iulie 2010

Cu deosebită considerație,
Cristea Ana Daniela

În memoria:

Unei persoane deosebite (mamei mele Sabina Cristea) care din păcate a plecat dintre noi pe parcursul redactării ultimei părți a prezentei teze, nemaiapucând astfel să participe la acest eveniment.

Cristea, Ana Daniela

Contribuții la realizarea și dezvoltarea unui nou concept de autorizare SAP pe bază de calificări și conectare fizică prin intermediul RFID

Teze de doctorat ale UPT, Seria 12, Nr. 1, Editura Politehnica, 2010, 144 pagini, 69 figuri, 10 tabele.

ISSN:1842-5208

ISBN:978-606-554-130-6

Cuvinte cheie: autorizare, calificative, acces controlat, SAP NetWeaver, securitate, pattern, autentificare, QBAC

Rezumat:
Lucrarea de față aduce contribuții în domeniul securității sistemelor informatice, respectiv în cadrul metodelor de acces controlat la resurse. În acest sens s-a dezvoltat o nouă metodă de acces la resurse fizice bazată pe calificative, precum și un nou pattern numit QBAC. Pattern-ul dezvoltat este recunoscut international fiind mentionat si de organizatia pattern-urilor pentru securitate. Faza de dezvoltare a metodei de acces controlat s-a prezentat folosind limbaje formale de genul UML și Data Mining, astfel încât această metodă să poată fi implementată folosind orice limbaj de programare. De asemenea s-a prezentat structura sistemului distribuit necesar implementării QBAC și modelul matematic rezultat. Implementarea la nivel de server s-a realizat cu ajutorul platformei SAP NetWeaver folosindu-se în acest sens diverse tehnologii SAP. În vederea integrării obiectelor protejate (mașini) în cadrul conceptului de autorizare dezvoltat s-a folosit PLC din familia Siemens, iar pentru comunicarea cu acestea s-a folosit tehnologia OPC.
Concluzia lucrării este că metodele de acces controlat ocupă un rol deosebit de important, iar metoda de acces controlat la resurse dezvoltată este o metodă necesară și fiabilă, ce trebuie însă dezvoltată de la varianta de test la o variantă care să poată fi implementată în viitor într-un sistem productiv.

CUPRINS

NOȚIUNI, ABREVIERI, ACRONIME

ABAP - Advanced Business Application Programming

ACL – Access Control List

ADS – Adobe Document Service

ALV – SAP List Viewer

AS – application Server

BAPI – Business Application Program Interfaces

BSP – Busienss Server Pages

CERT/CC – Computer Emergency Response Team / Coordination Center

CPU – Central Processing Unit

DAC – Discretionary Access Control

DCL – Data Control Language

DCS – Distributed Contrl Systems

DDL – Data Definition Language

DML – Data Manipulation Language

DoS – Denial of Service

DSD - Dynamic Separation of Duty

EPROM – Erasable Programmable Read Only Memory

ESS – Employee Self Service

FBD - Function block diagram

FG – Function Group

FM – Function Module

GPS - Global Positioning System

HMI – Human Machine Interface

I18N – Internationalization, multilanguage

IT- Information Technology

iView – integrated View

LAD - Ladder diagram

LAN – Local Area Network

LSO – Learning Solutions

LUW – Logical Unit of Work

MAC – Mandatory Access Control

MANDT – Data element, SAP client field

MVC - Model View Controller

OLTP – Online Transaction Processing

OOP – Object Oriented Programming

OPC - OLE for Process Control

OTR – Online Text Repository

PDF – Portable Document Format

PKI – Public Key Infrastructure

PLC - Programable Logic Controller

QBAC - Qualifications Based Access Control

RBAC – Role Based Access Control

RFC – Remote Function Call

RFID – Radio Frequency Identification

S7-PLCSIM – Step7 – PLC Simulation

S7-SCL – Step7 - Structured Control Language

SAP – nume firmă cu sediul central în Walldorf (Germania) fondată în 1972, SAP - (Systems Applications and Products)

SAP ERP – Entreprise Resource Planning

SAP GUI – Graphical User Interface

SAP HCM –SAP NetWeaver Human Capital Management

SAP HR – SAP NetWeaver Human Resources

SAP XI – SAP NetWeaver Excenge Infrastructure

SDN – SAP Developer Network

SO - Select Option

SOA - Sistem Oriented Arhitecture

SPAM – Expediere a mesajelor electronice nesolicitate

SPRAS – Data element, language key

SQL – Structured Query Language

SSD - Static Separation of Duty

SSO – Single Access Point

SSO – Single sign-on

STL - Statment list

TAN – Transaktionsnummer

UDDI – Universal Description Discovery and Integration

UI – User Interface

UML - Unifed Modeling Language

VPN – Virtual Private Network

WAN – Wide Area Network

WSDL - Web service Description Language

WYSIWYG - What You See Is What You Get

XML – Extensible Markup Language

LISTA DE TABELE

LISTA DE FIGURI

1. INTRODUCERE

1.1 Temă

Securitatea sistemelor informatice joacă un rol din ce în ce mai important, cu atât mai mult cu cât amenințările teroriste și "cybercrimele" sunt tot mai prezente. Indiferent dacă se vorbește despre securitatea informației, sau despre securitatea zonelor de acces, pentru realizarea acesteia dispunem de o serie de metode de acces controlat la resurse, metode descrise prin intermediul diferitelor pattern-uri.

Scopul formalizării și documentării pattern-urilor este dorința de a surprinde cât mai multe problematici legate de securitate, astfel încât posibilele soluții să poată fi universal înțelese, putând fi astfel folosite sau combinate pentru acoperirea anumitor brețe în acest domeniu. Există comunități și organizații care se ocupă cu design-ul pattern, precum și forumuri legate de aceast domeniu. Pattern-ul dezvoltat pentru descrierea metodei de acces controlat ce stă la baza temei prezentei teze, se încadrează în categoria pattern-urilor pentru securitatea sistemelor informatice. Acesta este menționat și de organizația pattern-urilor pentru securitate [135].

Așadar, zona de aplicabilitate a propunerilor prezentei teze este tocmai cea a metodelor de acces controlat la resurse, făcând totodată parte din importanta problematică a securității sistemelor informatice. Cuvântul securitate provine din cuvântul grecesc securus care înseamnă „fără grijă", acestei denumiri dându-i-se de-a lungul timpului o serie de definiții. În ceea ce privește securitatea sistemelor „fără grijă", descrie principial protejarea anumitor resurse împotriva posibililor atacatori, fiind așadar și tema pe care o vom dezbate în continuare, în primul rând din punctul de vedere al autorizării.

Prezenta lucrare reprezintă o sinteză a contribuțiilor aduse de autor. Rezultatele celor 21 lucrări acceptate spre publicare sau publicate în acest domeniu (**Anexa A1**) sunt sintetizate în acest material, iar cunoștințele din domeniul SAP au dus la materializarea a două cărți publicare în editura Springer. Totodată, o carte publicată în editura Mirton prezintă diverse posibilități de comunicare între un calculator și diferite aplicații externe, precum și modalitati de programare ale interfețelor de comunicare.

În ceea ce privește contribuțiile aduse pot spune că sunt contribuții actuale, marea parte a acestora au apărut în publicații cu vizibilitate (jurnale, conferințe indexate ISI sau BDI) iar patternul QBAC dezvoltat este menționat și de organizația pattern-urilor pentru securitate.

1.2 Obiective

Obiectivul principal al prezentei teze este obținerea unor rezultate noi, relevante pentru domeniul accesului controlat la resurse, cu precădere în autorizare.

Contribuțiile aduse prin lucrarea de față sunt în primul rând metoda de acces controlat dezvoltată, precum și pattern-ul QBAC realizat. Astfel, în familia patern-urilor pentru accesul controlat la resurse, QBAC ocupă un loc important în zona accesului la resurse fizice, prin care dreptul de acces la intrările unui obiect

protejat se face pe baza calificativelor sau aptitudinilor de care dispune un anumit subiect.

Obiectivele acestei teze sunt:

- definirea unei noi metode de acces controlat la resurse bazată pe calificative;
- crearea unui nou pattern;
- implementarea metodei create într-o versiune prototip;
- prezentarea fazei de dezvoltare într-un limbaj formal (astfel încât implementarea acesteia să poată fi realizată nu doar cu ajutorul tehnologiei SAP ci și cu ajutorul altor platforme sau limbaje de programare);
- prezentarea structurii sistemului distribuit necesar implementării metodei de acces propuse.

Necesitatea dezvoltării acestei metode de acces nu este una pur teoretică ci este o necesitate ce vine din nevoia concretă a sistemelor productive. După dezvoltări viitoare și retestări în vederea îndeplinirii cerințelor principale ale unui sistem distribuit, această metodă se va implementa în producție, câteva firme fiind interesate de o viitoare implementare.

1.3 Structură

S-a optat pentru o structură care să pornească de la prezentarea necesității metodelor de acces controlat la resurse, să continuie cu prezentarea contribuțiilor aduse în acest domeniu, cu prezentarea fazei de dezvoltare, urmând ca ulterior să se prezinte modul de implementare (software și hardware) a metodei de acces dezvoltate, precum și diversele tehnologii folosite în acest sens. Lucrarea este structurată pe un număr de 8 capitole. În cadrul acestora s-a acordat un spațiu extins capitolelor ce prezintă metoda de acces dezvoltată (prezentată prin intermediul limbajelor formale) și modul de implementare la nivel de server, restul capitolelor având o structură mai redusă.

Capitolul 2 are rolul de a introduce cititorul în problematica accesului controlat la resurse, de a prezenta diverse metode existente în acest domeniu precum și eventualele surse de atacuri și atacatori ai unui sistem. Totodată se scoate în evidență importanța și actualitatea securității, precum și faptul că aceasta reprezintă practic un sistem complex, ce necesită luarea în considerare a unui număr foarte mare de elemente și de problematici.

În **Capitolul 3** se prezintă metoda de acces controlat bazată pe calificative dezvoltată, alături de pattern-ul QBAC creat. Astfel, soluția dezvoltată va putea fi folosită de către un utilizator interesat de acest tip de problemă, sau de către oricine dorește să combine QBAC cu alte pattern-uri existente. Totodată, se prezintă faza de dezvoltare precum și algoritmii codare/decodare drept de acces. Toate acestea sunt descrise prin intermediul Data Mining și diverselor diagrame UML (ex. diagramă de clasă, diagramă de secvență, de activitate). Scopul descrierii fazei de dezvoltare prin intermediul limbajelor formale este dorința prezentării metodei dezvoltate astfel încât aceasta să poată fi implementată folosind orice limbaj de programare. Tot în acest capitol s-a scos în evidență necesitatea și avantajele avute în cazul folosirii pentru implementarea QBAC a unei platforme de integrare. Totodată, în cadrul acestui capitol s-a prezentat structura sistemului distribuit necesar implementării metodei de acces controlat dezvoltate, structura simplificată a sistemului rezultat

precum și modelarea matematică. Motivele prezentării sistemului într-o formă bloc simplificată este realizarea unui model matematic. Astfel, urmând prescrierile modelării matematice se vor lua în considerare doar elemente importante descrierii funcționalităților QBAC, fără a prezenta elementele care nu sunt importante din acest punct de vedere.

Capitolul 4 prezintă structura platformei de aplicații și integrare SAP NetWeaver precum și principalele argumente ale alegerii acesteia pentru implementarea pattern-ului QBAC. În următoarele subcapitole, acest extins capitol 4, prezintă diversele obiecte de dezvoltare realizate, precum și principalele tehnologii SAP utilizate (ex. Web Dynpro ABAP, SAP NetWeaver portal, SAP ERP HCM, formularele Adobe). Pentru fiecare tehnologie s-a scos în evidență avantajele și dezavantaje acestora, precum și motivele pentru care s-a ales respectiva tehnologie pentru implementarea QBAC. Pe lângă toate acestea, s-au prezentat și principalele elemente de limbaj ABAP utilizate dar și modul în care platforma SAP NetWeaver ajută în realizarea aplicațiilor multilinguale, plecând de la interfețele cu utilizatorul și ajungând până în zona datelor stocate în bazele de date.

Capitolul 5 începe cu descrierea diverselor metode de autentificare existente și continuă cu prezentarea tehnologiei RFID folosită ca și metodă de autentificare pentru implementarea metodei de acces dezvoltate. Schimbarea metodei de autentificare se va putea realiza fără a influența politica de autorizare dezvoltată. În acest mod se va putea oferi fiecărui client o metodă de acces controlat la resurse bazată pe calificative, metodă în care tipul de autentificare să poată fi ales în funcție de dorințele clientului.

Totodată, acest capitol scoate în evidență premisele alegerii acestei tehnologii pentru proiectul de față, alături de avantaje, dezavantaje și drepturi ce trebuiesc respectate pentru persoanele ce folosesc o astfel de tehnologie.

Capitolul 6 aduce în prim plan tehnologia OPC și modalitățile de programare ale unui PLC din familia Siemens. În primul subcapitol se prezintă diversele avantaje ale folosirii aplicației soft Step7 și se scoate în evidență modul de programare al PLC-ului, precum și principalele limbaje de programare utilizate în acest sens.

În următorul subcapitol se trece la prezentarea tehnologiei OPC cu cele două componente principale (OPC server și OPC client), scoțând în evidență motivele alegerii acestei tehnologii în vederea realizării legăturii dintre server (platforma SAP NetWeaver) și PLC. Totodată, acest capitol prezintă și modul de implementare al comunicării dintre server și PLC în vederea obținerii datelor necesare de la server sau în vederea înscrierii anumitor valori la nivelul serverului.

Capitolul 7 prezintă tipul PLC-ului folosit pentru integrarea mașinilor în conceptul de autorizare, prezintă diversele module ale PLC-ului ales, prezintă standul de test realizat precum și rezultatele experimentale. Totodată în cadrul acestui capitol se analizează sistemul distribuit (obținut în urma implementării QBAC) din punct de vedere al funcționalităților ce trebuiesc îndeplinite de către un astfel de sistem.

Pe lângă toate acestea, capitolul de față prezintă motivele alegerii controller-elor de tip PLC pentru scopul implementării QBAC, precum și avantaje ale acestora și modalități de comunicare dintre un PLC și o stație de lucru.

Capitolul 8 prezintă concluziile finale, principalele contribuții aduse de autor precum și dezvoltări pentru viitor necesare tranziției de la faza de model la faza în care metoda de acces prezentată se va putea implemeta productiv.

2. ACCESUL CONTROLAT LA RESURSE INFORMAȚIONALE ȘI FIZICE

În societatea zilelor noastre atentatele realizate asupra sistemelor informatice sunt amenințări concrete iar termeni de genul: hacker, SPAM, phising, fraude electronice, cyberterorism sunt denumiri pentru care fiecare dintre noi le cunoaște semnificația. În acesată societate, în care protejarea informației și securizarea zonelor de acces sunt necesități cotidiene, accesul controlat la resurse joacă un rol din ce în ce mai important, acesta asigurând că o entitate (ex. persoană, sistem) poate accesa doar acele informații, sau doar acele resurse fizice pentru care are autorizarea corespunzătoare.

Scopul capitolului de față este tocmai acela de a scoate în evidență necesitatea accesului controlat la resurse (informaționale și fizice), de a prezenta diferite modele de acces controlat existente la ora actuală în acest domeniu precum și modelele care au fost reconsiderate pentru obținerea unei noi soluții.

2.1 Necesitatea accesului controlat la resurse informaționale și fizice

În lupta competițională industrială dezvoltarea de sisteme sigure, care să raspundă nevoilor de securitate și să ofere soluții din ce în ce mai complexe este o necesitate a unei societăți moderne, în continuă evoluție.

Metodele de acces la resurse stau la baza unor sisteme care se utilizează zi de zi. De exemplu, sub ecranul de login al Windows-ului, după introducerea parolei și parcurgerea secțiunii de autentificare, se declanșează un mecanism de autorizare având la bază modelul Extended Authorization cu ACL. Atunci când ne logăm la portalul firmei se acordă drept de acces pe baza rolurilor pe care user-ul le are atașate, folosind modelul RBAC. Un alt exemplu al folosiri metodelor de acces controlat poate fi situația în care se accesează zonele securizate dintr-o instituție. După parcurgerea procesului de autentificare se acordă (de cele mai multe ori) drept de acces bazat pe roluri - RBAC.

Accesul controlat la resurse se face pe baza autentificării și a autorizării, trebuind să ținem cont totodată și de faptul că folosirea unui concept de autorizare solid, sau a unei metode de autentificare cu un grad ridicat de siguranță nu implică automat obținerea unui sistem sigur. Toate acestea trebuie îmbinate cu folosirea unor metode de securitate adecvate [1], cu o programare sigură, care să înlăture punctele slabe (vulnerabilitățile) și să țină la distanță eventualele atacuri. Așadar, accesul controlat nu poate fi oferit ca o funcționalitate de sine stătătoare, trebuind combinată cu alte servicii pentru a obține securitatea necesară.

În lucrarea IBM „White Paper" [2], se afirmă:

"Security isn't just a product, and it isn't just a service. It's a condition that is expected to be embedded in the process of creating value."

Atunci când se vorbește despre securitate, și în mod particular despre accesul controlat la resurse, trebuie să se țină cont de câteva elemente principale, ca de exemplu [3], [4]:

- **Identificarea și Autentificarea** (Identification and Authentication I&A) – este procesul prin care se identifică și se verifică dacă o entitate este cea care declară că este.
- **Autorizarea** (Authorization) – este procesul prin care după autentificarea entității i se acordă acesteia doar acele drepturi (pe baza politicilor de securitate) prin care aceasta să poată avea acces la resursele necesare pentru a-și îndeplini sarcinile.
- **Trasabilitatea** (Accountability) – este procesul prin care ne asigurăm că activitatea unei entități într-un sistem poate fi reconstituită.
- **Integritatea** (Integrity) – exprimă faptul că informația vehiculată nu a suferit modificări în timpul transportului, sau că informația stocată nu a fost modificată de către entități neautorizate.
- **Confidențialitatea** (Confidentiality) – exprimă faptul că accesul la resurse trebuie să se acorde doar acelor entități care sunt autorizate să aibă acest drept.

În ultima perioadă, s-au înmulțit incidentele în care datele personale ale clienților și/sau angajaților au ajuns la persoane străine, fiind folosite în diverse scopuri, ca de exemplu: SPAM, infectare cu „viermi" informatici, etc. ajungându-se până la fraude electronice de mari dimensiuni, [5], [6], [7], [8], sau accesare neautorizată a anumitor zone cu acces limitat. De asemenea, experții în domeniu [9], [10] au tras un semnal de alarmă în ceea ce privește creșterea cazurilor în care prin diferite tehnici (ex. phishing, pharming, cititoare de card si camere montate la ATM-urile bancilor) se urmărește obținerea datelor de autentificare, astfel încât să se abuzeze de autorizarea unei entități sau situații în care explorându-se slabiciunile unui sistem, au fost realizate atacuri [11] asupra instituțiilor guvernamentale, sau asupra marilor furnizori de servicii (ex. energie, comunicații).

În Fig. 2.1 se prezină doar câteva dintre posibilele atacuri ce pot fi realizate asupra unui sistem sau asupra unei rețele [12], [13].

Internetul a cunoscut o dezvoltare explozivă, majoritatea firmelor mutându-și o parte din activitate pe internet. Astfel, se permite angajaților acces de la distanță la rețele securizate, posibilitatea de a comanda prin intermediul internetului anumite procese industriale, posibilitatea de a accesa anumite date vitale prin intermediul serviciilor web, etc. Sistemele închise și simpliste în care accesul la resurse se realizează local nu mai este un scenariu des întâlnit. Una din necesitățile pe care globalizarea le-a adus cu ea este accesul la resursele unei corporații din toate centrele acesteia distribuite în diferite colțuri ale lumii, necesitatea realizări sistemelor industriale distribuite, utilizarea de protocoale standard și a rețelelor din ce în ce mai complexe. Acest lucru a dus și la extinderea sistemelor industriale către sisteme distribuite complexe. În acest tip de sisteme, pentru asigurarea securității și

protejarea resurselor (ex. informații, echipamente, software, servicii, persoane) trebuie să se realizeze o colaborare între personalul IT și inginerii de sistem.

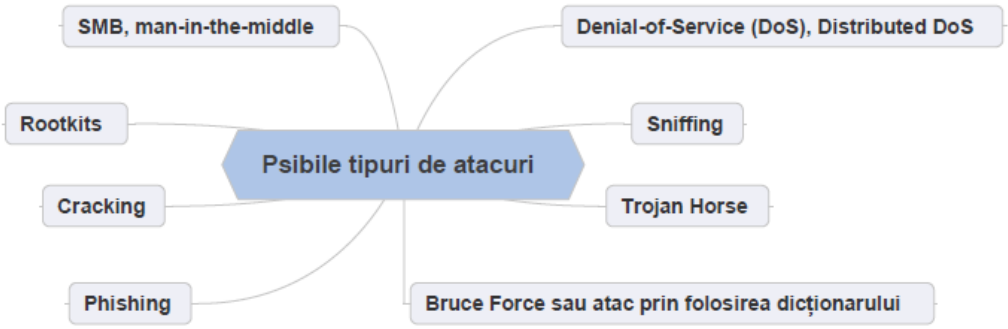


Fig. 2.1 Posibile tipuri de atacuri asupra unui sistem sau rețea

Dacă se aruncă o simplă privire asupra domeniului internet, care este prezent în marea majoritate a activităților (ex. sisteme industriale, comerț) un studiu realizat de CERT/CC [14] arată o creștere îngrijorătoare a numărului de vulnerabilități înregistrate de-a lungul anilor (1998 - 2006), Fig. 2.2. În acest studiu se prezintă vulnerabilitățile catalogate pe baza rapoartelor provenite din surse publice.

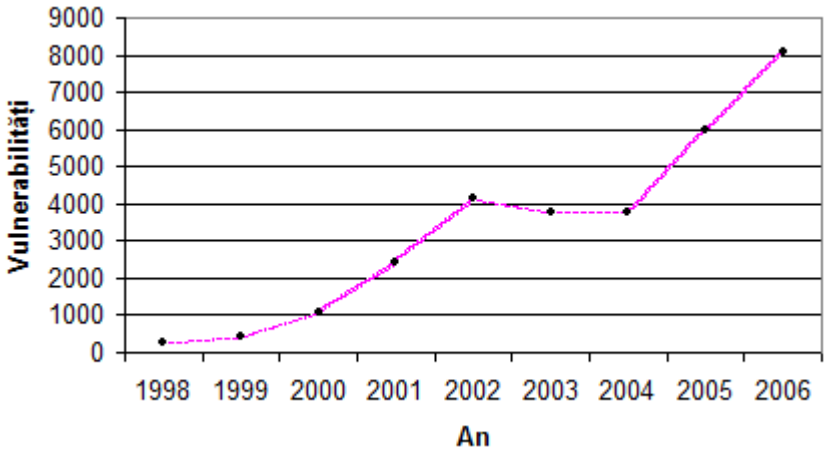


Fig. 2.2 Statistcă CERT/CC privind numărul de vulnerabilități raportate

Privind doar aceste date legate de vulnerabilitatea unei mici părți dintr-un sistem modern de acces controlat, protejarea datelor confidențiale ale unei instituții, protejarea datelor clienților și angajaților, accesul controlat la resurse informaționale și resurse fizice este o problemă a zilelor noastre, breșele de securitate în acest domeniu provocând pagube însemnate.

Conform unei statistici oferite de CSI [15] avand ca și intervievați 144 de organizații care au dorit să ofere informații în acest sens, pagubele suferite din cauza diferitelor probleme de securitate (ex. viruși, acces neautorizat la sisteme) pe

parcursul anului 2008 este in medie de 288.618 $ pe intervievat. Totodată, trebuie să se ia în considerare și faptul că multe atacuri nu sunt facute publice din cauza dorinței companiilor de a nu-si pierde reputația. În lucrarea [16] se fac o serie de precizări privind modul de interpretare al statisticilor legate de „computer crime".

În Fig. 2.3 se prezintă posibile surse de atacuri pentru un sistem informațional [5], [10].

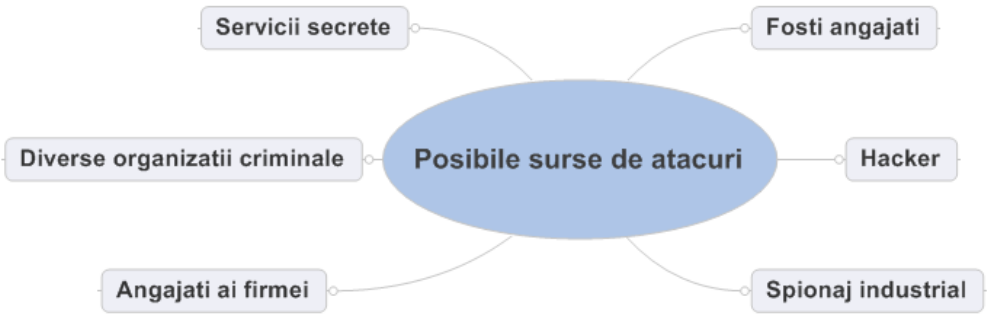


Fig. 2.3 Posibile surse de atacuri

Corespunzător unui studiu privind motivația angajaților în „computer crime" [17], se arată că pentru organizațiile cu mai mult de 250 de angajați, 57% din incidentele de securitate se referă tocmai la proprii angajați. Așadar, securizarea unui sistem nu trebuie să fie neglijată nici în cazul în care funcționalitățile respective se realizează pentru proprii angajați.

În acest context, metodele de acces controlat la resurse informaționale și fizice joacă un rol special deoarece ele, alături de alte metode (ex. Criptografie), reprezintă garantarea faptului că o entitate poate avea acces doar la acele obiecte protejate pentru care are autorizarea adecvată.

2.2 Pattern-uri pentru accesul controlat la resurse

Este unanim recunoscut că design-ul pattern își are originile în arhitectură, plecând de la conceptul vizionar al lui Christopher Alexander [18]. Acesta a folosit pentru prima data noțiunea de pattern, oferind o definiție simplă, dar totodata la obiect:

"Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem in such a way that you can use this solution a million times over, without doing it the same way twice "[18]

Erich Gamma, Richard Helm, Raplh Johnnson și John Vlissider sunt considerați ca cei care au revoluționat pattern-ul în programarea orientată obiect. În cartea lor [19] au descris design pattern-ul ca:

„How we describe design pattern? Graphical notations, while important and useful, aren't sufficient. They simply capture the end product of the design process as relationschip between classes and objects"

Ulterior s-au creat o multitudine de pattern-uri într-o largă gamă de categorii. De exemplu, doar în categorie a pattern-urilor pentru securitate s-au creat ca și subcategorii [20]: acces controlat la resurse, acces controlat pentru sisteme de operare, Identificare & Autentificare (I&A), arhitecturi Firewall, aplicații internet sigure, etc.

Modelele pentru accesul controlat la resurse se pot încadra principial în trei categorii de bază : DAC, MAC și RBAC [21].

- **DAC** - printr-un astfel de model se înțelege o modalitate de acces controlat în care nu se dispune de un administrator, fiecare utilizator putând să își administreze singur obiectele de care dispune, astfel încât să ofere drepturi pentru acestea. În acest caz accesul la resurse se face pe baza unei matrici de acces dispunând de trei actori principali: (***S, O, A***), unde **S** – set de subiecte, **O** – set de obiecte, **A** – matricea de acces.

- **MAC** - printr-un astfel de model se înțelege o modalitate de acces controlat la resurse, în care se dispune de un administrator ce se ocupă cu acordarea drepturilor (ex. citire, scriere), pe care utilizatorii le au în sistem.

- **RBAC** – acest tip de model de acces controlat este bazat pe roluri. Astfel o entitate dispune de un user, user care are asignate anumite roluri pe baza cărora se determină dreptul acesteia în sistem. Totodata se dispune de administratori pentru crearea și managementul userilor și rolurilor. Fundamental se dispune de trei actori principali (***S, R, P***) unde **S** – subject, **R** – role, **P** – permisiune. Acest model este considerat un model non-discretional ce are multe variante (ex. modelul de bază RBAC, modelul ierarhic RBAC, modelul **SSD**, modelul **DSD**).

Generalizând, celelate modele existente pot fi încadrate în unul din modelele prezentate, diferențele dintre acestea fiind legate de setul de reguli folosit pentru definirea drepturilor și permisiunilor oferite într-un anumit sistem.

Modelele pentru acces controlat la resurse sunt sintetizate în colecții de pattern-uri, de cele mai multe ori reprezentate sub forma diagramelor de clasă UML. Prezentarea acestora sub forma unor pattern-uri ofera o serie de ***avantaje***, ca de exemplu [22], [23], [24]:

- posibilitatea de a învăța din experiența celorlați;

- reutilizarea soluțiilor existente;

- posibilitatea de a combina soluții existente pentru obținerea unor soluții mai complexe;

- posibilitatea de a crea variante ale soluțiilor propuse;

- posibilitatea de a prezenta soluții prin intermediul unui limbaj formal - independent de platforma în care acestea au fost implementate.

Câteva dintre ***cerințele*** pe care trebuie să le îndeplinească un sistem pentru acces controlat la resurse, conform pattern-ului Access Control Requirements [20], sunt:

- respingere a eventualelor accesări neautorizate;
- permiterea accesărilor autorizate;
- limitarea pagubelor în cazul unui acces neautorizat;
- folosirea politicilor de autorizare.

O clasificare generală a pattern-urilor în funție de frecvența folosirii lor, poate fi făcută în așa numitele [25]:

- **Good practices patterns *(*Well know patterns)** – întâlnite în cel puțin trei sisteme reale.
- **Useful solutions patterns –** prezintă soluții ce nu au fost folosite sau soluții ce au fost implemetate o singură data dar a căror idei pot fi refolosite.

Astfel, în cazul în care ne aflăm în fața unei probleme legate de accesul controlat la resurse se pot studia soluțiile existente (fie well know patterns fie useful solutions patterns) astfel încât să se refolosească o soluție existentă, să se refolosească anumite părți componente, sau să se combine soluțiile existente astfel încât să se obțină o soluție a problemei, beneficiind de experiența sistemelor deja existente.

Avându-se în vedere că problema principală a prezentei teze de doctorat este aceea de a acorda angajaților unei firme posibilitatea de a accesa intrările anumitor mașini (obiecte protejate) în funcție de calificativele (aptitudinile) de care aceștia dispun, s-au studiat modelele de acces controlat la resurse existente. Datorită faptului că nici unul dintre modele nu îndeplinește în totalitate cerințele problemei menționate, plecând de la ideile de bază a câtorva pattern-uri pentru acces controlat la resurse s-au conceput și dezvoltat elemente proprii, astfel încât să se ofere acces la intrările obiectelor protejate în funcție de calificative.

În zona pattern-urilor pentru acces controlat la resurse putem avea atât acces la resurse informaționale, cât și acces la resurse fizice.

În Tabelul 2.1 se prezintă pattern-urile pentru acces la resurse informaționale care au fost reconsiderate în vederea dezvoltarea noii soluții propuse. Pattern-urile prezentate în acest tabel sunt descrise în [26 - 29].

O comparație a acestora bazată pe anumite criterii ca de exemplu: confidențialitate, costuri de implementare, securitate, autorizare, autentificare se poate găsi la [30].

Nume Pattern	Utilizări cunoscute	Acces controlat la resurse
Extended Authorization	Bază pentru sistemele de acces controlat al marilor produse comerciale ca de exemplu Unix, Windows, Oracle	Bazat pe drepturi
Session	RBAC, MBAC, UNIX ftp, telnet services	Bazat pe sesiuni pentru a îndeplini principiul: "least privilege"
Role Based Access Control RBAC	.NET, J2EE, SAP NetWeaver, Oracle DBMS, SELinux, Microsoft SQL server	Bazat pe rolurile pe care un user le are
Metadata Based Access Control MBAC	.NET	Bazat pe proprietățile subiectelor și a obiectelor

Tabel 2.1 Pattern-uri pentru accesul controlat la resurse informaționale

Datorită faptului că pattern-ul „authorization" este baza multor altor metode pentru accesul controlat la resurse, se prezintă în continuare succint, principalele sale componente, Fig. 2.4 (adaptat din [20], [21]).

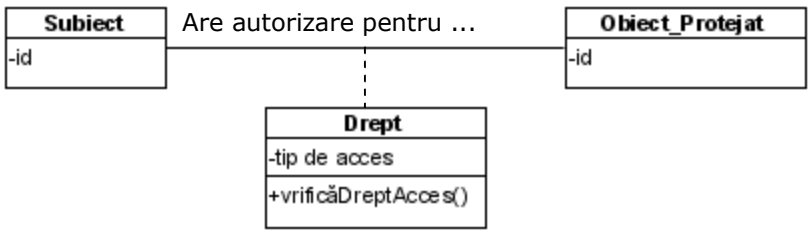


Fig. 2.4 Structura pattern-ului authorization

Astfel, un subiect poate accesa un obiect protejat pe baza drepturilor pe care acesta le are (ex. citire, scriere). În acest mod se poate verifica dacă subiectul respectiv dispune de dreptul de a accesa respectivul obiect protejat. Implementarea acestuia este de cele mai multe ori realizată cu ajutorul ACL.

Pe lângă pattern-urile pentru accesul controlat la resurse informaționale prezentate, un alt pattern care a fost reconsiderat pentru definitivarea soluției propuse este pattern-ul pentru accesul controlat la resurse fizice: „Access Control to Physical Structures „ [31]. Acest pattern folosește modelul RBAC (pentru determinarea dreptului unei entități de a accesa o structură fizică) împreună cu pattern-urile: Alarm Monitoring, Relays, Time Shedule, pentru a controla sistemul atunci când apar evenimente. Prin intermediul acestui pattern se oferă o soluție în care se acordă subiecților posibilitatea de a accesa intrările unei clădiri în funcție de rolurile de care aceștia dispun.

Concluzii:

În cadrul acestui capitol:

- s-a realizat un studiu critic asupra problematicilor generale ale accesului controlat la resurse;
- s-a scos în evidență necesitatea accesului controlat la resurse (folosind statistici și grafice);
- s-au analizat diversele surse de atacuri ale unui sistem informatic, precum și posibili atacatori ai acestuia;
- s-au prezentat modelele de bază ale accesului controlat la resurse;
- s-au prezentat pattern-urile care ne-au inspirat în obținerea noii soluții care se va propune în cadrul capitolului următor.

3. CONTRIBUȚII LA ACCESUL CONTROLAT LA RESURSE FIZICE

Capitolul de față este structurat pe trei părți. Prima parte este dedicată prezentării metodei de acces controlat la resurse fizice, ce reprezintă contribuția autorului, metodă de acces formalizată prin intermediul unui nou pattern numit QBAC [32].

În a doua parte a acestui capitol se va prezenta faza de devoltare, astfel încât acest concept să poată fi implementat folosind orice limbaj de programare. Faza de dezvoltare este urmată de prezentarea algoritmilor de codare și decodare a dreptului de acces.

Ultima parte este dedicată prezentării sistemului distribuit necesar implementării QBAC precum și modelarea matematică a acestuia. Totodată, se scoate în evidență necesitatea folosirii unei platforme de integrare (la nivel de server) si necesitatea folosirii unui sistem distribuit (la nivel fizic).

3.1 QBAC – Qualification Based Access Control Pattern

Pattern-ul propus și prezentat în acest paragraf se încadrează in categoria „userful solutions pattern", fiind un pattern pentru acces controlat la resurse fizice non-discretional, a cărui implementare se pretează foarte bine pentru un sistem distribuit. Soluția descrisă de acest pattern a fost implementată ca și prototip o singură dată, pentru scopul acestui proiect, dar ideea prezentată poate fi refolosită și în alte situații similare.

După cum s-a precizat, pattern-urile care au ajutat la definitivarea acestei soluții sunt: Extended Authorization, Session, RBAC, MBAC și Access Control to Physical Structures. În afara reconsiderării ideilor de bază a acestora, a fost nevoie de adăugarea unor ***elemente adiționale*** necesare îndeplinirii ***cerințelor*** noastre:

1. Subiectelor li se va acorda drept de access pe bază de calificative. Calificativele au proprietatea că pot expira după o anumită perioada de timp. Pentru reobținerea acestora subiectul trebuie să participe la școlarizare.

2. În afara sesiunilor necesare implementării principiului "last privilege" a fost necesară adăugarea unor funcționalități suplimentare pentru motive de siguranță la nivelul obiectelor protejate;

3. Obiectele protejate și calificativele dispun de anumite atribute.

Pentru a descrie un pattern se propun mai multe formate [20], [33] prin intermediul cărora se poate realiza o descriere a acestuia într-o formă corespunzătoare (ușor de înțeles) și totodată se oferă posibilitatea de a furniza detaliile necesare implementării acestuia. Pe lângă proprietățile principale: **context** (care specifică sitația în care apare nevoia respectivului pattern), **problema** (care

descrie problemele pe care respectivul pattern le rezolvă și **soluția** (care ofera o soluție a problemei), dispunem și de o serie de alte proprietăți mai mult sau mai puțin obligatorii ca de exemplu: nume, intenție, consecințe, exemplu din lumea reală, implementare, forțare, structură, utilizări cunoscute și variante, Fig. 3.1.

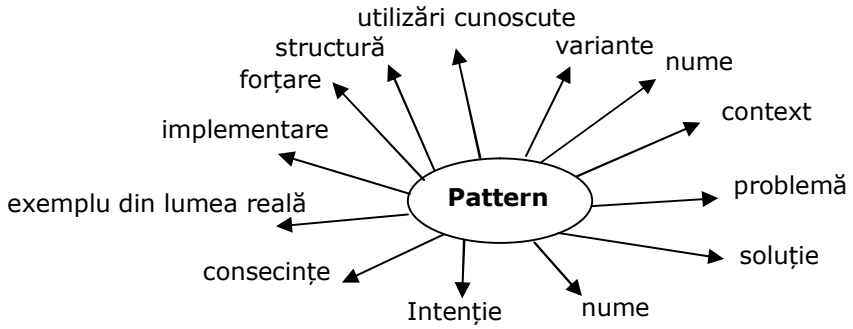


Fig. 3.1 Posibile elemente pentru descrierea unui pattern pentru accesul controlat la resurse

Pentru pattern-ul QBAC am optat sa folosim: numele pattern-ului, context, problemă, forțare, intenție, exemplu din lumea reală, soluție, structură, consecințe și utilizări cunoscute.

Numele pattern-ului: QBAC (**Q**ualification **B**ased **A**ccess **C**ontrol)

Intenție: Accesul controlat la intrările anumitor obiecte protejate (mașini), acces bazat pe calificativele pe care un subiect (angajat) le are**.**

Context: Orice set de obiecte protejate la care este nevoie de acces controlat la anumite intrări ale lor și unde subiectele pot fi clasificate în concordanță cu aptitudinile lor, pot dispune de calificative multiple și pot participa la școlarzare pentru a-și îmbunatăți calificativele.

Exemplu din lumea reală: În Fig. 3.2 se prezintă o sesiune de logare a unui subiect la unul dintre obiectele protejate.

Pașii parcurși din momentul în care subiectul dorește să se logheze și până în momentul în care i se acordă acestuia dreptul de access pe baza calificativelor sunt:

- **Pas 1** – Subiectul se loghează la unul din obiectele protejate la care acesta dorește să acceseze anumite intrări.
- **Pas 2** – ID-ul subiectului este citit de către PLC de pe cardul de identificare RFID.
- **Pas 3 & 4** – ID-ul obiectului protejat (la care se dorește drept de acces) și ID-ul subiectului sunt transmise mediului de execuție (în cazul nostru platforma SAP NetWeaver).

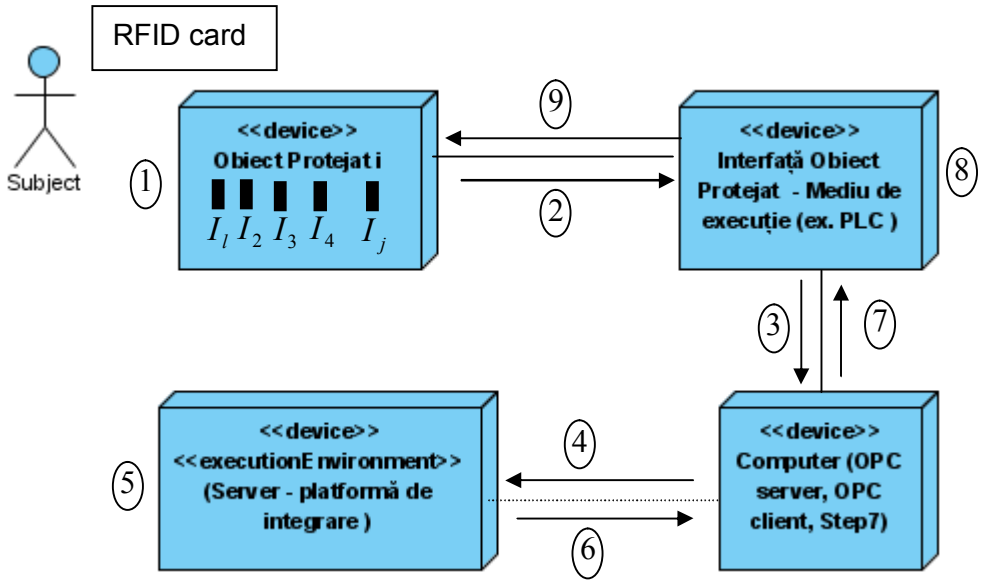


Fig. 3.2 Scenariu de logare

- **Pas 5** – După operația de autentificare se calculează dreptul respectivului subiect de a accesa intrările obiectului protejat la care dorește să se logheze.

- **Pas 6** – Datele sunt trimise la OPC client codate într-un întreg. Dreptul de acces este trimis codat într-un întreg pentru a nu încărca rețeaua de comunicare în cazul comunicării prin Web Service și pentru a ușura comunicarea în cazul folosirii directe prin intermediul ActiveX.

- **Pas 7** – Datele sunt trimise la PLC pentru a crea comanda.

- **Pas 8** – Dreptul de acces este decodat și se crează comenzile adecvate.

- **Pas 9** – Comenzile sunt transmise obiectului protejat, oferind posibilitatea subiectului de a accesa doar acele intrări pentru care dispune de calificativele corespunzătoare.

Problemă: Trebuie să se definească un mod în care să se controleze accesul la intrarile obiectelor protejate pe bază de calificative, astfel încât să se poată respinge cererile de acces neautorizate.

Forțare: Întregul sistem trebuie sa fie dinamic, astfel încât să permită adăugarea de noi obiecte protejate și de asemenea să permită adăugarea de noi

intrări unui obiect protejat existent. Aceste schimbări trebuie sa fie ușor implementabile fără a necesita efort deosebit din partea administratorului.

Pentru a mări siguranța în funcționare trebuie îndeplinite două condiții inițiale: un subiect poate accesa la un moment dat doar intrările unui singur obiect protejat, iar un obiect protejat va putea fi deservit la un moment dat doar de un singur subiect, care dispune de calificative corespunzatoare

Trebuie să se implementeze principiul „least privilege" pentru a se oferi doar privilegiile de care un subiect are nevoie pentru a-și îndeplini sarcinile.

Soluție: În Fig. 3.3 este prezentat un model de bază al Qualifications Based Access Control

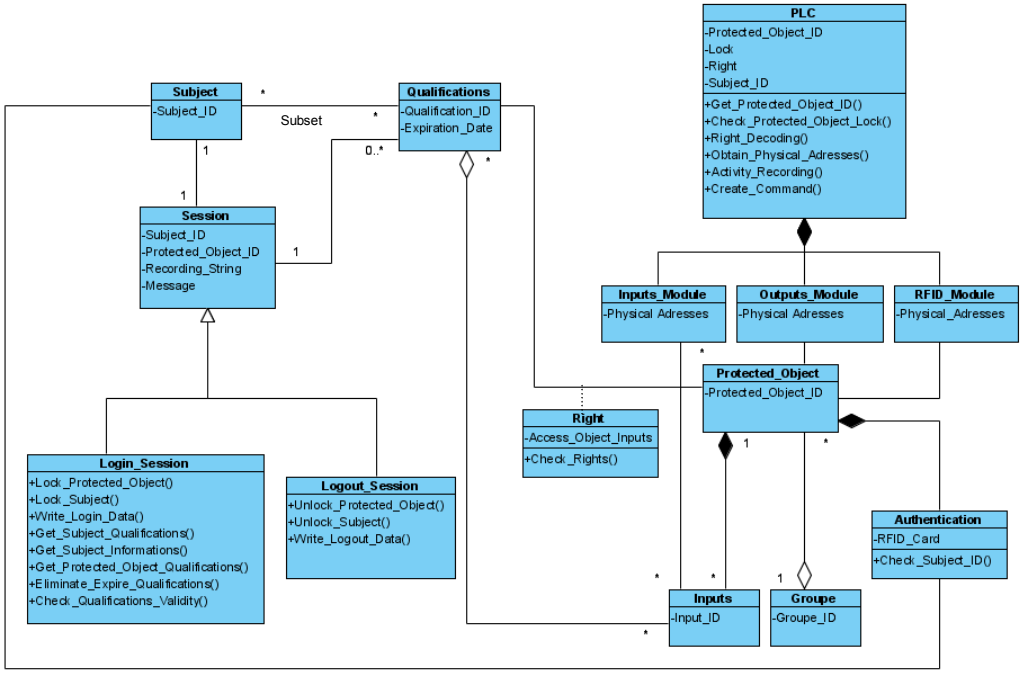


Fig 3.3 Modelul de bază QBAC

Clasa Subiect descrie un subiect ce dorește să acceseze anumite intrări ale obiectului protejat.

Clasa Protected_Object reprezintă resursa a cărei intrări trebuie să se protejeze. Deoarece putem avea un număr foarte mare de obiecte protejate s-a realizat o organizare în grupe. În acest mod fiecare grupă poate avea orice numar de obiecte protejate și, la rândul lui, fiecare obiect protejat poate avea orice număr de intrări.

Clasa Qualifications deconectează conexiunea directă dintre obiectul protejat (resursă) și subiect, reprezentând calificativele subiectului. Subiecții dispun de calificative corespunzător cărora primesc drepturi de accesare a intrărilor unui obiect protejat. De exemplu Subiectul x poate dispune de calificativul "Installer" pentru un obiect protejat. Corespunzător acestui calificativ va putea să acceseze intrările care au fost asignate de către administrator acestui calificativ.

Din punct de vedere al unui administrator, relația ***calificativ – drept de acces*** poate fi schematizată ca în Fig. 3.4.

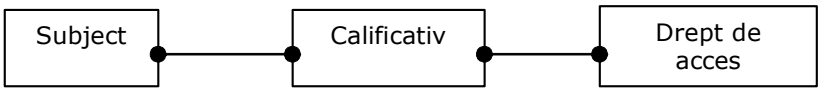


Fig. 3.4 Relație Subiect – Calificativ – Drept de acces

Un calificativ are proprietatea că poate expira după o perioadă de timp, calificativele expirate trebuind să fie eliminate.

Pentru ca un subiect să poată reobține un calificativ care a expirat, sau să își mărească gradul de autorizare, trebuie să participe la școlarizare. În cazul calificativelor multiple se vor înlătura redundanțele care pot să apară.

Oricare Subiect poate interacționa cu sistemul (server) prin intermediul unei sesiuni (sesiune de logare sau sesiune de delogare) de unde își activează calificativele. Sesiunile permit să se implementeze principiul „least privilege”. Orice sesiune de logare oferă doar atatea privilegii de câte are nevoie un subiect pentru a accesa intrările corespunzătoare aptitudinilor sale.

O sesiune de delogare va avea ca și rezultat zero calificative, dar va realiza operații de genul:

- înregistrare a anumitor valori (ex. a timpului cât un subiect a fost logat la respectivul obiect, activitatea acestuia);
- deblocare subiect, deblocare obiect protejat.

Pentru a nu da posibilitatea unui subiect să se logheze la mai multe obiecte protejate, în cadrul unei sesiuni de logare se va bloca subiectul respectiv, urmând a-l debloca în urma unei sesiuni de delogare. De asemenea se va bloca și obiectul protejat la care subiectul a realizat operația de logare, astfel încât să ne asigurăm că la un moment dat acesta poate fi deservit doar de un singur subiect.

Clasa **Authentication** reprezintă metoda de autentificare prin intermediul căreia se realizează autentificarea subiecților la nivelul obiectelor protejate. Schimbarea metodei de autentificare de la carduri RFID la o altă metodă, nu va influența politica de autorizare.

Consecințe: Deoarece prin intermediul calificativelor se oferă drepturi subiecților, nu este necesar să se asigneze intrările direct subiecților. În momentul când apar noi intrări pentru un obiect protejat, această modificare va fi foarte ușor realizată printr-o simplă asignare a intrărilor adăugate calificativelor existente pentru respectivul obiect. Automat, toți subiecții care dispun de calificativele respective vor putea deservi intrările noi adăugate, fără a fi nevoie de modificări adiționale. O excepție apare doar atunci când intrările respective nu pot fi asignate calificativelor existente, în acest caz trebuind să se realizeze unul sau mai multe calificative precum și cursurile aferente, astfel încât subiecții să poată participa la

școlarizare și să obțină calificativele conform cărora vor avea dreptul să acceze noile intrări.

Prin intermediul *sesiunilor* s-a realizat principiul "least privilege", astfel încât să se ofere unui subiect doar atâtea drepturi de câte are nevoie pentru a accesa intrările obiectului protejat pe care dorește să le deservească.

Prin intermediul *mecanismelor de blocare* a subiecților și a obiectelor protejate se asigură că:

- un subiect nu accesează mai multe obiecte în acelasi timp;
- un obiect protejat nu poate fi deservit de mai mulți subiecți în acelasi timp.

Prin intermediul codării realizate nu se trimit stringuri care conțin adresele fizice ale intrărilor la care respectivul subiect are drept de access, ci se trimite un numar întreg, ușor de decodat (reobținând după decodare adresele fizice ale intrărilor pe care un subiect are dreptul să le acceseze). De asemenea folosirea mecanismului de codare va fi de un real ajutor în procesul de administrare.

Folosirea calificativelor în loc de roluri este avantajoasă și din punctul de vedere al resurselor umane putând realiza pe baza profilelor diferite rapoarte de genul:

- de ce calificative are nevoie un subiect ca să poată să obțină un anumit job;
- ce calificative îi mai trebuie unui subiect să poată să obțină de exemplu postul de instalator pentru grupul de mașini X sau pentru mașina X.Y;
- susține procesul continuu de învățare al angajaților (subiecților).

O altă consecință este aceea că se poate extinde pattern-ul QBAC cu pattern-ul RBAC (obținând un QBAC extins) atunci când se oferă soluții moderne de învățare a angajaților, sau atunci când se oferă subiecților posibilitatea de a rezerva un curs la care doresc să participe (Fig. 3.5).

O varianta a pattern-ului propus se poate folosi pentru a acorda drepturi de acces a personalului în anumite încăperi, în funcție de calificativele acestuia.

Utilizări cunoscute: Metoda de acces controlat concepută, formalizată prin intermediul acestui pattern, se află în fază de testare la firma NWCON Technology Consulting GmbH - Germania.

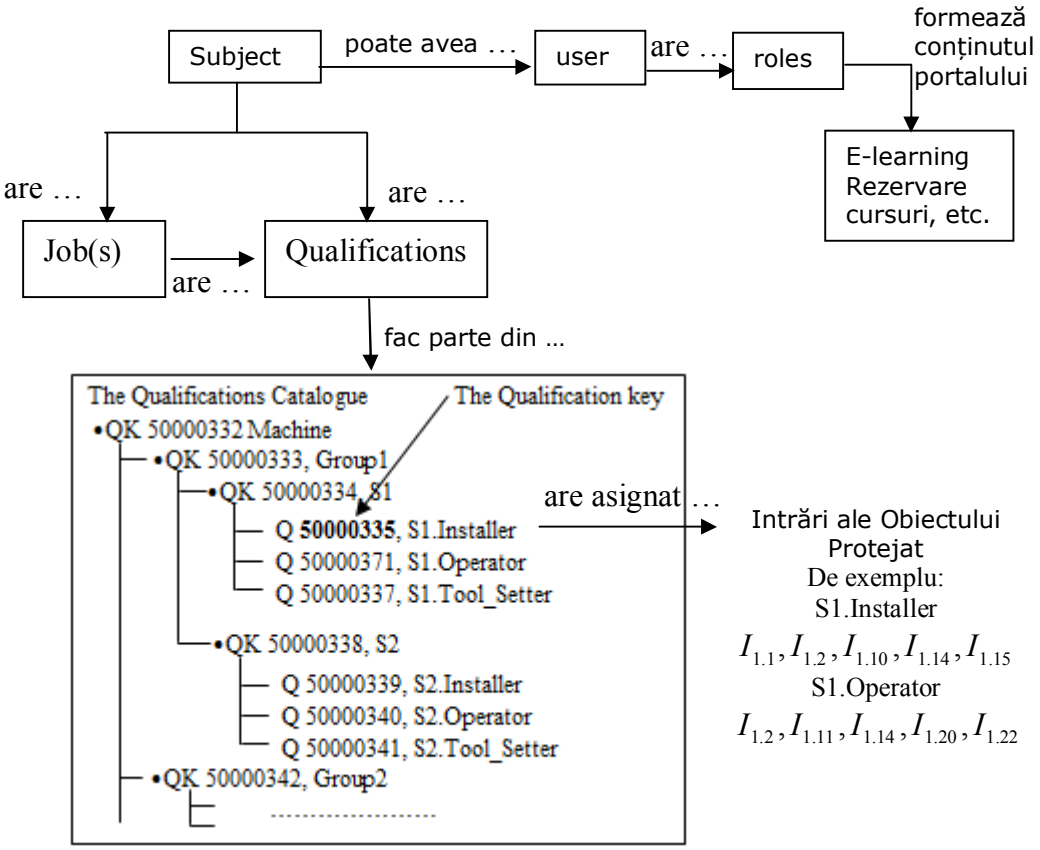


Fig. 3.5 Combinare QBAC cu RBAC

3.2 Prezentarea fazei de dezvoltare

Sub formă abstractizată se poate reprezenta sistemul de autorizare ca în Fig. 3.6, unde subiectului (angajatului) i se permite accesarea intrărilor unui sistem (mașină) din cadrul unei mulțimi de sisteme pe baza calificativelor.

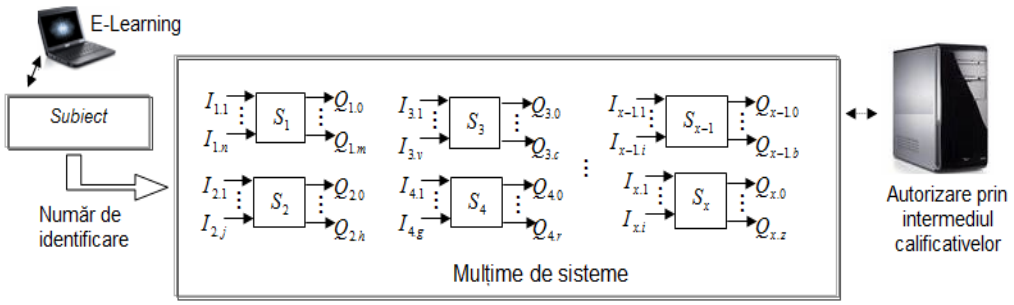


Fig. 3.6 Sructura abstractizată a sistemului rezultat

În continuare se va prezenta faza de dezvoltare a metodei de acces controlat dezvoltată, folosind în cea mai mare parte diagramelel UML îmbinate cu elemente de tipul Data Mining.

Diagramele UML sunt folosite curent în dezvoltarea de software, precum și în faza de dezvoltare a diferitelor proiecte, oferind astfel un limbaj adecvat modelării, care poate fi înțeles de către toți participanții la proiect. Acest tip de limbaj se folosește cu succes nu doar în sistemele software, ci și în alte medii, ca de exemplu în domeniul bancar sau diferite fluxuri economice.

La ora actuală există o mare varietate de produse software cu care se pot crea diagrame UML. Putem vorbi aici atât despre produse ce se pot downloada și instala gratuit cât și despre produse complexe ce necesită licență. Principalul avantaj al produselor din a doua categorie este acela că ne oferă unelte adiționale ca de exemplu:

- generare de cod;
- automatizarea mapării între diverse obiecte (model de obiecte, model de date) și baza de date relațională.

Un exemplu de astfel de produs soft pentru realizarea diagramelor UML este Visual Paradigm for UML [34]. Acesta a fost folosit pentru crearea diagramelor necesare în cadrul acestui proiect.

Diagrame UML cel mai frecvent utilizate sunt [35], [36], [37], [38], [39]:

- **Diagrame de clase** (class diagram) – sunt unele dintre cele mai folosite diagrame UML, oferind posibilitatea de a descrie o largă varietate de modele. După cum am putut observa în cadrul acestui capitol, inclusiv pentru prezentarea pattern-urilor s-au folosit diagrame de clase UML.
- **Diagrame de secvență** (sequence diagram) – sunt acele diagrame cu ajutorul cărora se poate reprezenta comportamentul unui sistem sau a unei componente.
- **Diagrame de obiecte** (object diagram) - sunt acele diagrame care prezintă obiectele împreună cu relațiile acestora.
- **Diagrame de activitate** (activity diagram) - acestea prezintă fluxuri de control pornind de la o stare de start urmate de anumite tranzacții (ce pot avea inclusiv ramificații asemănătoare celor din schemele logice) și încheind cu o stare finală.
- **Diagrame contexte de utilizare** (use case diagram) - sunt folosite de cele mai multe ori pentru a reda necesitățile unui sistem, prezentând totodată interacținea dintre un actor și sistemul respectiv.
- **Diagrame de componente** (component diagram) – sunt folosite pentru a se prezenta structura unui sistem, explicitând totodată modul în care diferitele componente ale acestuia sunt conectate.

Data Mining [40], [41], [42] va fi folosit în cadrul tezei pentru extragerea din larga cantitate de date ce pot fi create doar a acelor date importante QBAC. De exemplu, datele subiecților se pot stoca într-o singură tabelă sau se poate crea o bază de date ce să conțină până la sute de tabele. Folosind Data Mining se scot în evidență (extrage) doar acele informații fără de care logica QBAC nu poate funcționa.

Adesea, se folosește limbajul natural în comunicarea cerințelor unui sistem informatic, limbajul UML în faza de dezvoltare al respectivului sistem, Data Mining în realizarea patter-urilor pentru extragerea datelor iar diversele limbaje de programare (ex. ABAP, Java, Visual Basic) pentru implementarea soluțiilor. Urmând această logică, s-au prezentat cerințele sistemului de acces controlat la resurse, în cadrul acestui subcapitol se va prezenta faza de dezvoltare, urmând ca pe parcursul capitolelor următoare să se prezinte modul de imlementare, folosind diversele limbaje de programare și tehnologii.

3.2.1 Faza de dezvoltare la nivel de server

3.2.1.1 Prezentarea elementelor componente

Pentru a obține un sistem flexibil la schimbări, logic, ușor de administrat, sigur în funcționare, ce elimină pe cât posibil redundanțele și care să codeze autorizarea vehiculată între mediul de execuție (server) și mulțimea de sisteme, s-a împărțit întregul concept de acces controlat pe patru nivele.

Părțile componente sistemului sunt (Fig. 3.7):

- calificativele (Qalifications);
- numerele personale din HR (Identification Numbers);
- useri plus roluri (portal users, portal roles) folosite pentru procesul de învățare, pentru administratorul nivelului celui mai de jos al autorizarii, etc.;
- id-ul sistemelor și intrările acestora (bază de date relațională).

După cum s-a precizat întregul sistem trebuie să fie dinamic, astfel încât la apariția unor noi intrării la unul dintre sisteme această modificare să poată fi ușor procesată la nivelul cel mai de jos al autorizării, fără a se propaga spre nivelele superioare. Administratorul acestui nivel va atașa noua intrare unuia dintre calificativele existente pentru respectivul sistem și, automat toți subiecții ce au asignat calificativul respectiv vor putea accesa intrarea respectivă. Doar în cazuri exceptionale, când noua intrare nu poate fi asignată unuia din calificativele existente pentru sistemul respectiv, administratorul care se ocupă de nivelul cel mai de jos al autorizării va trebui să ia legatura cu administratorul ce se ocupă cu Learning Solutions pentru a crea un nou calificativ pentru sistemul respectiv, precum și un nou curs necesar noului calificativ. Astfel subiecții vor putea participa la școlarizare, vor putea obține calificativul respectiv, dispunând astfel de autorizarea necesară accesări intrării respective. În acest mod pentru fiecare sistem se crează un număr de calificative iar fiecarui calificativ i se asignează un număr de intrări. După participarea la școlarizare și îndeplinirea sarcinilor ce se cer, fiecare subiect va primi

unul sau mai multe calificative, având astfel automat autorizarea de a accesa intrările sistemelor corespunzatoare.

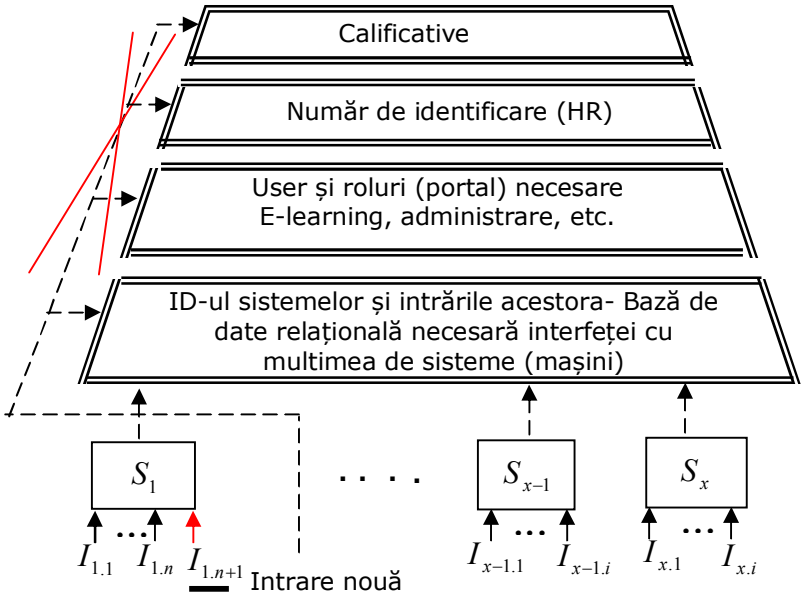


Fig. 3.7 Părți componente sistemului

3.2.1.2 Structura bazei de date

Plecând de la premisa că, crearea datelor subiecților, a calificativelor, a user-ilor și a rol-urilor se poate realiza cu unul dintre uneltele puse la dispoziție de mediul în care se implementează acest concept, ceea ce trebuie să realizăm este interfața dintre acestea și mulțimea de sisteme. Pentru aceasta se va dezvolta o bază de date relațională (Fig. 3.8), punctând totodata și locul în care trebuie să se realizeze legatura cu celelalte componente. În cazul în care sistemul informatic nu oferă suport pentru crearea calificativelor și a datelor subiectelor, această baza de date se poate foarte ușor extinde prin intermediul unui număr de tabele adiționale necesare păstrării informațiilor respective.

După cum se poate observa prin intermediul tabelei ***Connection_with_Learning_Solutions*** am simbolizat legatura cu *tabela* unde se vor găsi obiectele de tip ***Qualifications*** (Calificative) generate în urma populării cu date a catalogului de calificative corespunzător mulțimii de sisteme.

Un sistem informatic ce ofera suport de genul **Learning Solutions** va genera o mulțime de tabele relaționate unde vor fi păstrate toate elemenentele legate de calificative, cursuri, denumiri, etc. Legatura cu acestea se va realiza prin intermediul *tabelei* unde este stocată cheia unica a obiectului de tip ***Qualification***, în cazul nostru notata cu OBJID. În cazul în care sistemul în care se implementează acest concept de autorizare nu ne pune la dispozitie un astfel de modul, putem extinde baza de date cu un numar de tabele ce variază în funcție de dorința fiecăruia de a avea o cantitate mai mare sau mai mică de informații.

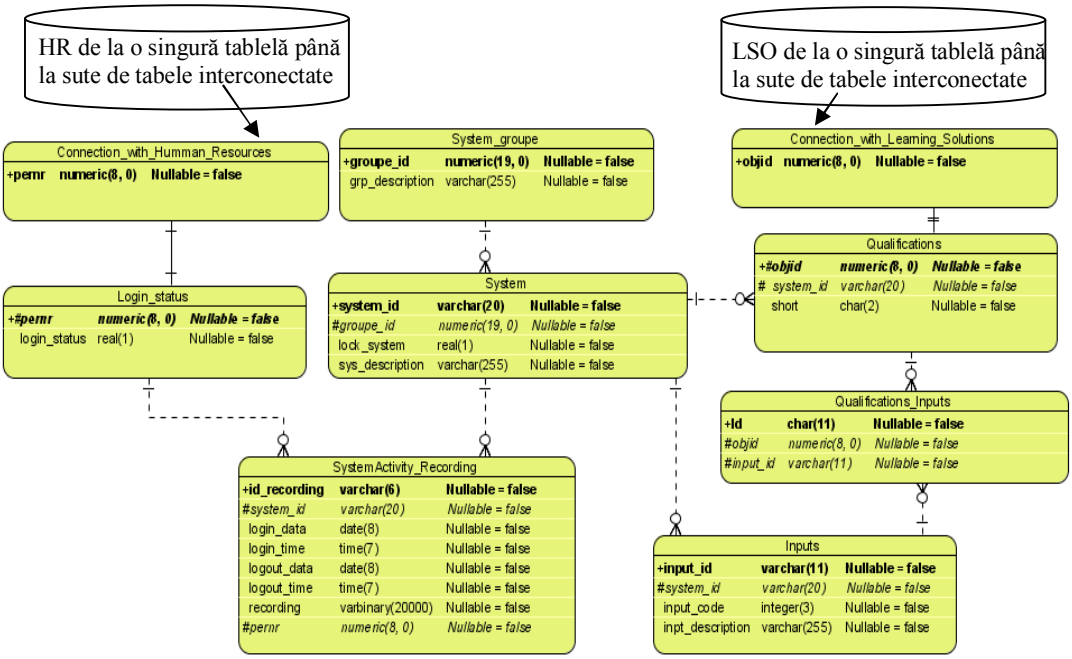


Fig. 3.8 Structura bazei de date

Important pentru scopul conceptului de autorizare prezentat sunt ***Qualifications*** împreună cu data de început a valabilității acestuia, data de expirare a valabilității acestuia și descrierea aferentă:

Attributes = (objid, bigin_data, end_data, description)

Pentru o mai bună organizare se includ aceste calificative în cataloage, într-un sistem complex, putând avea calificative nu doar pentru mulțimea de obiecte protejate ci și pentru alte domenii ca de exemplu: limbi străine sau Microsoft Office (Word, Excel, PoWer Point).

Cursurile sunt importante în cadrul metodei de acces controlat prezentată doar din punct de vedere al oferirii posibilității de participare la școlarizare pentru obtinerea calificativelor, neavand un rol direct în calculul autorizării.

Pentru a crea calificativele putem realiza obiecte de tip Q, fiecare avand o cheie unică generată. De exemplu calificativul ***Instalator*** pentru sistemul S1 poate fi un obiect Q de forma: 50000335. Toate calificativele ce au aceași apartenență se recomandă a se crea în același catalog de calificative, iar ca și scală pentru obținerea acestora putem folosi varianta cea mai simpla YES/NO.

Prin intermediul tablei ***Connection_with_Human_Resources*** am simbolizat legatura cu tabela unde se vor găsi datele referitoare la subiect, date ca de exemplu: nume, adresă, informații legate de educație, calificativele de care aceasta dispune, instituția în care lucrează, etc. Așadar, putem avea de la o singură tabelă Atribute = (pernr, nume, prenume, telefon, calificative) până la sute de tabele interconectate. De exemplu folosind SAP ERP HCM (partea de HR) putem avea până la 1000 de tabele pentru păstrarea datelor (infotype 0000 – 0999).

Legatura cu aceste date se va realiza prin intermediul numărului de identificare ce reprezinta cheia unica a acestor informații, în cazul nostru notată cu PERNR.

Pentru cazul în care nu ni se ofera un modul pentru crearea cursurilor putem proceda la fel ca și în cazul Calificativelor, creând catalogul de cursuri împreună cu cursurile aferente. De exemplu un curs poate fi un obiect de tip C, care are o cheie unica generată, în același format ca și calificativele.

Pentru o mai bună definire am împărțit întreaga mulțime de sisteme în grupe de sisteme, fiecare grupă putâd avea ***n*** sisteme, iar fiecare sistem putând avea ***m*** intrări. Pentru a evita posibilitatea în care un subiect se poate loga simultan la mai multe sisteme am creat posibilitatea de a bloca respectivul subiect folosind în acest scop coloana ***login_status*** a tabelei ***Login_status***. În cazul în care aceast câmp conține valoarea 0 subiectul este deblocat, el putandu-se oricând loga la unul din sistemele existente, iar când această valoare devine 1 subiectul respectiv va fi blocat. Acestui subiect nu i se va mai oferi autorizarea de a accesa intrările unui alt sistem doar după delogare de la sistemul pentru care a fost blocat.

În mod similar am procedat și pentru mulțimea de sisteme. Un sistem (mașină) din cadrul acestei mulțimi va fi blocat atunci cand un subiect a realizat o operație de logare și va fi deblocat după ce acel subiect se va deloga. În acest fel ne asigurăm că un sistem este deservit la un moment dat de un singur subiect.

Pentru a avea informații legate de activitatea subiecților după logarea lor la un sistem: ora și data când s-a realizat logarea, ora și data când s-a realizat delogarea, sistemul la care s-a realizat logarea/delogarea, numele subiectului ce deservește intrările respectivului sistem am creat tabela ***System_Activity_Recording***.

3.2.1.3 Fluxul necesar determinării drepturilor subiecților

În Fig. 3.9 se prezintă prin intermediul unei diagrame UML de activitate modul în care s-a implementat algoritmul de determinare al dreptului de acces pe care îl are un subiect la unul dintre sistemele pe care acesta dorește să îl deservească.

După cum putem observa calificativele (obținute de catre subiecți în urma unui proces de învățare) stau la baza acestui sistem de autorizare. Ca și principală proprietate a unui calificativ este data de expirare. Acesta este motivul pentru care înainte de a calcula dreptul unui anumit subiect la unul din sisteme trebuie să excludem calificativele expirate. Vom informa totodată subiectul respectiv de faptul că nivelul scăzut de autorizare de care dispune momentan se datorește calificativului sau calificativelor expirate. Totodată, înainte cu 20 de zile de data de expirare se va trimite un mesaj de avertizare către acel subiect, aceasta având astfel timp să își refacă școlarizarea înainte de a-și pierde nivelul de autorizare curent.

În cazul calificativelor multiple va trebui să se înlăture eventualele redundanțe care pot apărea, iar apoi vom coda întreaga autorizare într-un întreg obținând astfel dreptul de acces al subiectului respectiv. La nivelul controller-elor (ex. PLC) va urma să se facă apoi decodarea acestui întreg (reprezentând dreptul de acces), pentru a reobține adresele fizice ale intrărilor pe care subiectul are dretul să le deservească.

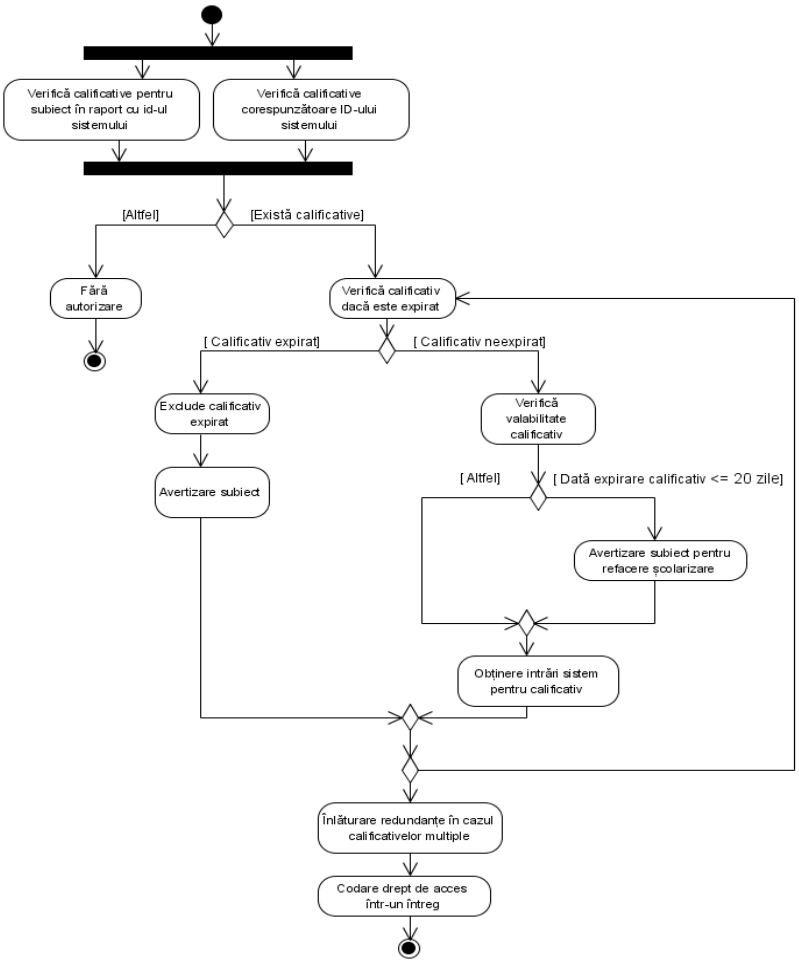


Fig. 3.9 Diagrama de activitate UML pentru implementarea algoritmului necesar determinării dreptului de acces al unui subiect la unul din cele n sisteme

Eliminarea redundanțelor în cazul calificativelor multiple ale unui subiect x pentru un sistem este necesară deoarece pot apărea situații în care prin calificative multiple vom obține drept de acces pentru intrări identice, Tabel 3.1.

Așadar, ***modelul de acces controlat*** prezentat constă din următoarele componente de bază:

- un set de calificative;
- un set de subiecte;
- un set de permisiuni de care dispun subiecții pentru accesarea intrărilor sistemelor.

Ex. Subiect x are Drept_de_acces=(S1.Instaler, S1.Operator)	
Drept de acces cu redundanțe - S1	Drept de acces fără redundanțe - S1
Drept de acces S1.Installer $I_{1.1}, I_{1.2}, I_{1.10}, I_{1.14}, I_{1.15}$ S1.Operator $I_{1.2}, I_{1.11}, I_{1.14}, I_{1.20}, I_{1.22}$	Drept de acces $I_{1.1}, I_{1.2}, I_{1.10}, I_{1.14}, I_{1.15}, I_{1.11}, I_{1.20}, I_{1.22}$ intrări redundante eliminate $I_{1.2}, I_{1.14}$

Tabel 3.1 Exemplu de situație de eliminare a redundanțelor

Pentru fiecare subiect calificactivul (calificativele) activ este acela (acelea) pe care subiectul îl (le) utilizează momentan (3.1):

$$AQ(s:subject) = \{\text{Calificativ (Calificative) activ pentru subiectul } s\} \quad (3.1)$$

Fiecare subiect poate să fie autorizat să folosească unul sau mai multe calificative (asignate lui de către administrator) (3.2):

$$QA(s:subject) = \{\text{Calificativ (Calificative) autorizat pentru subiectul } s\} \quad (3.2)$$

Fiecare calificativ poate fi autorizat să ofere drept de acces la una sau mai multe intrări ale unui sistem (3.3):

$$INP(q:qualification) = \{\text{intrări autorizate (care pot fi accesate) pentru calificativul q}\} \quad (3.3)$$

Fiecare subiect poate accesa intrările obiectelor protejate. Predicatul ***access(s,i)*** este adevărat dacă subiectul s poate accesa intrarea i la momentul current, altfel acest predicat este fals (3.4):

$$access(s:subject, i:input) = \text{TRUE if subiectul s poate accesa intrarea I} \quad (3.4)$$

Pentru accesul controlat pe bază de calificative (QBAC) se impun următoarele ***reguli de bază***:

- ***Asignarea calificativelor***: Un subiect poate accesa intrările unui sistem doar dacă subiectul are asignat cel puțin un calificativ (3.5):

$$\forall s:subject, i:inputs, (access(s,i) \Rightarrow AQ(s) \neq \phi) \quad (3.5)$$

- ***Autorizarea calificativelor***: Calificativele active ale unui subiect trebuie să fie autorizate (asignate de către administrator) respectivului subiect (3.6):

$$\forall s : subject, (AQ(s) \subseteq QA(s)) \qquad (3.6)$$

- ***Autorizarea intrărilor***: Un subiect poate accesa intrările unui sistem doar dacă intrările sunt autorizate (asignate de către administrator) pentru calificativul (Calificativele) activ al subiectului (3.7):

$$\forall s : subject, i : input, (access(s,i) \Rightarrow i \in INP(AQ(s))) \qquad (3.7)$$

3.2.1.4 Crearea datelor necesare procesului de învățare, administrarea acestora

În Fig. 3.10 se prezintă diagrama contextelor de utilizare UML, care oferă informații legate de cerințele minimale ale nivelului celui mai de sus al autorizării, partea legată de calificative și modalitățile de creare și organizare al procesului de învățare.

Administratorul acestui nivel va fi responsabil nu doar cu crearea cataloagelor de calificative și a calificativelor aferente, ci și cu crearea cataloagelor de cursuri, a cursurilor, cu pregătirea sistemului, astfel încât subiecții să aibă acces atât la participarea la școlarizări desfașurate în săli de clasă, cât și la participarea la E-learning. Totodata, în cazul în care apare un nou subiect (de exemplu o persoană este nou angajată în cadrul firmei), aceasta dispune deja de o serie de cunoștințe, administratorul acestui nivel va putea astfel să îi asigneze subiectului anumite calificative fără ca aceasta să mai participe la școlarizare. Același lucru este valabil și pentru subiecții care nu sunt angajați ai firmei respective - de exemplu un angajat aparținând unei alte instituții va vizita firma pentru o perioadă de timp.

După cum s-a specificat, în cazul în care pentru implementarea acestui concept se va folosi o platformă de integrare, administrarea acestui nivel se va face cu ajutorul uneltelor puse la dispoziție de către acea platformă. Doar în cazul în care nu se folosește o platformă de integrare ce oferă astfel de funcționalități va trebui să se realizeze și aplicația prin intermediul căruia să se ofere apoi unui administrator posiblitatea de a realiza aceste operații.

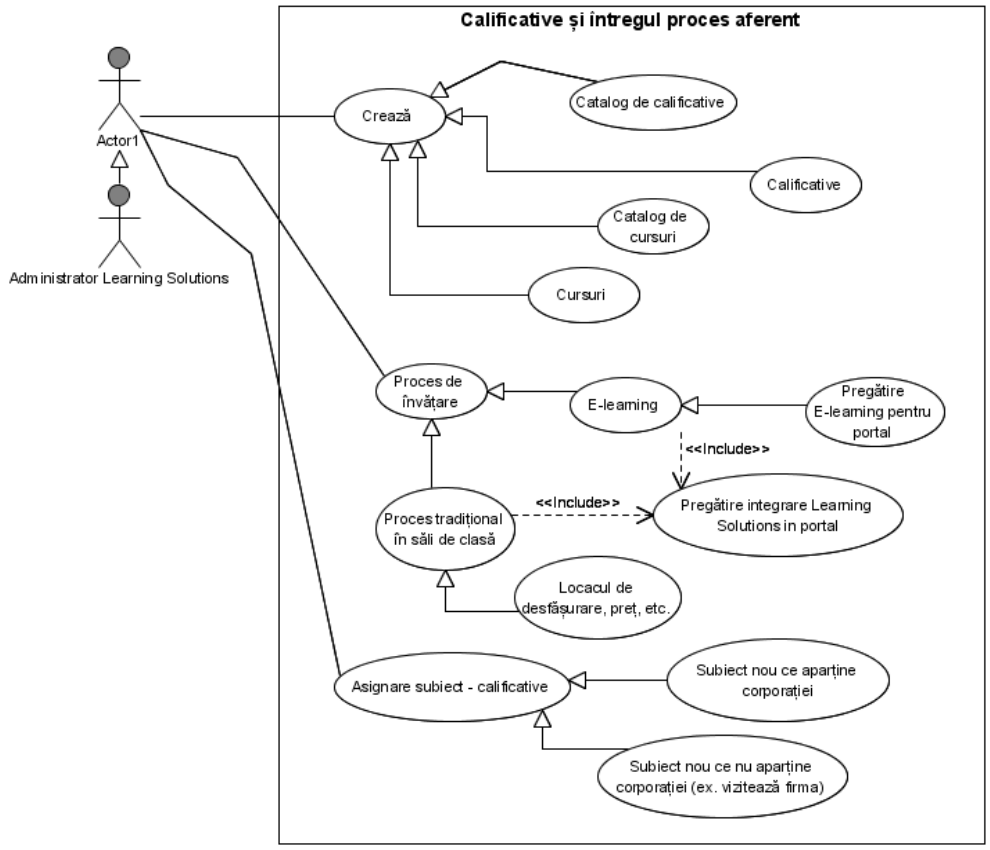


Fig. 3.10 Diagrama contextelor de utilizare – proces de învățare

3.2.1.5 Crearea și administrarea datelor angajaților

În cazul în care sistemul în care se implementează acest concept nu ne pune la dispoziție o modalitate de creare și administrare a acestor date, se poate crea propria aplicație de administrare. În mod minimal aceasta trebuie să ofere suport pentru urmatoarele operații, Fig. 3.11.

În acest mod se vor crea toate datele de care avem nevoie pentru a avea informații legate de fiecare subiect. De exemplu datele de contact pot fi: date legate de locul ocupat în firmă, date personale, elemente de contact, privilegii de călătorie.

Datele personale pot fi la randul lor: nume, prenume, inițiale, titlu, locul nașterii, naționalitate, limbă, etc.

Dintre toate datele rolul cel mai important îl are cheia unică: ***Numărul de identificare*** (Identification Number) – ***PERNR*** alături de nume, prenume, număr de telefon și calificative asignate lui.

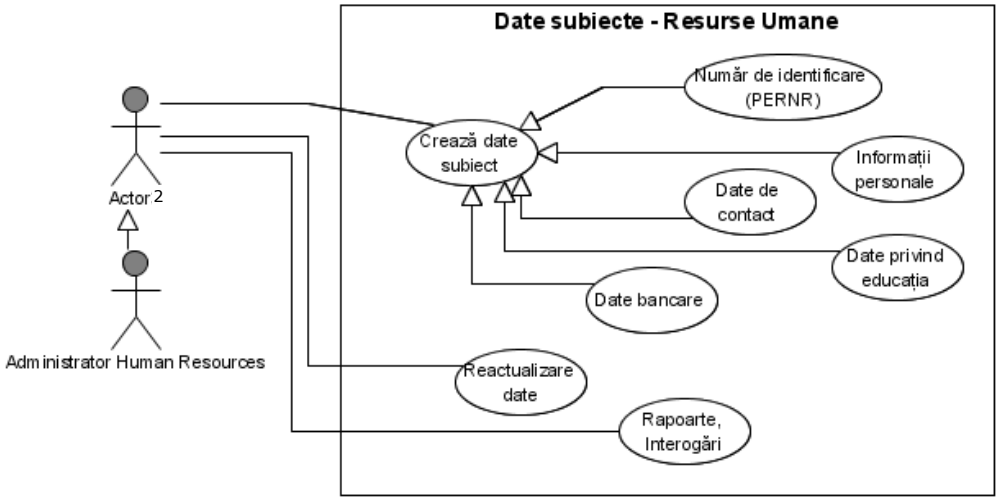


Fig. 3.11 Diagrama contextelor de utilizare – proces dedicat subiecților

3.2.1.6 Crearea și administrarea nivelului de interfață cu mulțimea de sisteme

Această aplicație de administrare a nivelului de legătură cu mulțimea de obiecte protejate trebuie să se realizeze și în cazul în care se va folosi o platformă de integrare, deoarece această interfațare nu există standard în nici o platformă.

În Fig. 3.12 se prezintă diagrama UML a contextelor de utilizare.

Principalele ***funcționalități*** pe care o astfel de aplicație de administrare trebuie să le ofere sunt:

- ***căutare avansată*** avand ca și atribute de căutare Căutare1=(system_id, objid, input_id, lock_system), Căutare2=(pernr, login_data, logout_data) și Căutare3=(pernr, login_status). Rezultatul căutării va fi afișat în tabele precum și în format PDF;

- ***selectare***, ce are ca și atribute de căutare în vederea obținerii rapoartelor rapide (ce vizează toți angajații): Select1=(pernr, login_data, first_name, last_name) și Select2=(system_id, lock_system, groupe_id, sys_description);

- ***importare***: a) importarea cheii subiecților pe care dorim sa îi introducem în conceptul de autorizare Import1=(pernr, login_status=default). Valoarea default va fi 0, deoarece inițial subiectul este neblocat;
  b) importarea intrărilor sistemelor și a denumirii acestora dintr-un fișier text Import2=(input_id, input_description, input_code=generate) În acest caz valorile din atributul input_code se generează fiind necesare conceptului de codare;

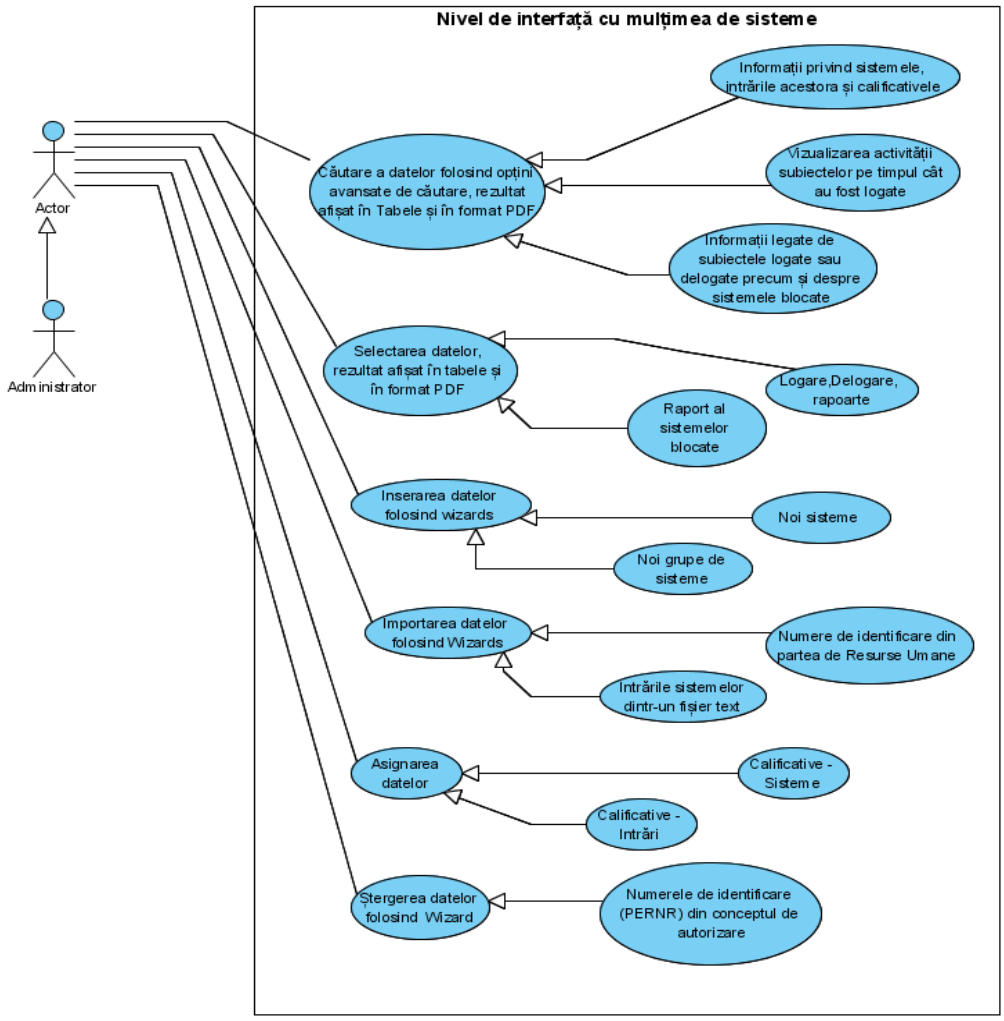


Fig. 3.12 Diagrama contextelor de utilizare – proces dedicat interfeței cu sistemele

- posibilitate de ***inserare*** a grupelor de sisteme și a sistemelor în baza de date creată: Insert1=(groupe_id, grp_description) și Insert2=(system_id, groupe_id, lock_system=default, sys_description). Lock_system este inclus automat ca și default = 0, sistemul fiind initial neblocat;

- posibilitate administratorului de ***asignare*** la calificative a intrărilor sistemelor pentru care se ofera drept de acces și posibilitate de asignare pentru fiecare sistem a calificativelor corespunzatoare;

- ***ștergerea*** cheilor subiecților din conceptul de autorizare în cazul în care aceștia nu mai au nevoie de accesarea intrărilor sistemelor, sau în cazul în care părăsesc firma Delete=(pernr, login_status). Cheile respective și

informațiile aferente angajaților vor rămâne stocate doar în partea de resurse umane.

3.2.1.7 Crearea și administrarea datelor din portal

Pentru administrarea datelor în portal avem nevoie de urmatoarele operații de bază, Fig. 3.13.

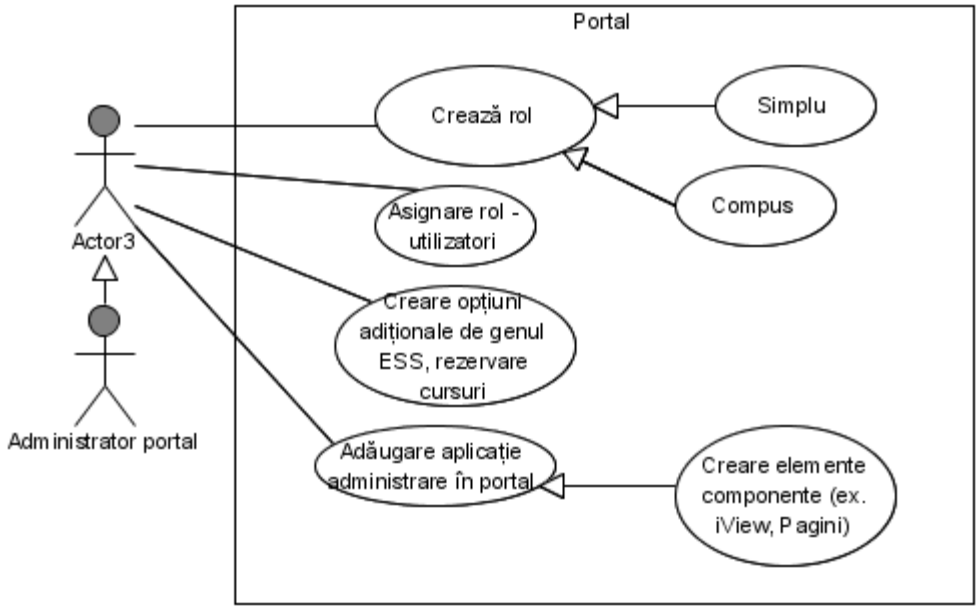


Fig. 3.13 Diagrama contextelor de utilizare – portal, operații de bază

Pentru realizarea acestor operații se folosesc uneltele puse la dispoziție de către mediul în care se implementează acest concept. Pentru cazul în care nu dispunem de un astfel de mediu se poate crea o aplicație minimală prin care să se ofere drept de acces pe bază de roluri la aplicația de administrare al interfeței cu mulțimea de sisteme.

3.2.2 Faza de dezvoltare la nivelul controller (PLC)

Pentru realizarea comenzilor către obiectele protejate (mașini) și primirea informațiilor de la acestea este nevoie de o serie de algoritmi si pași necesari, astfel încât să se poată:

- citi numărul de identificare de la subiect;
- determina id-ul sistemului (mașinii) la care se dorește logarea;
- decide dacă se face logare sau delogare la acea mașină;

- trimite datele la server;
- decoda dreptul angajatului obținând adresele fizice la care acesta are drept de acces;
- forma comenzile;
- face înregistrarea activității subiectului la mașina la care s-a logat, etc.

Diagrama de secvențe din Fig. 3.14 prezintă principalele acțiuni ce trebuiesc realizate în acest sens.

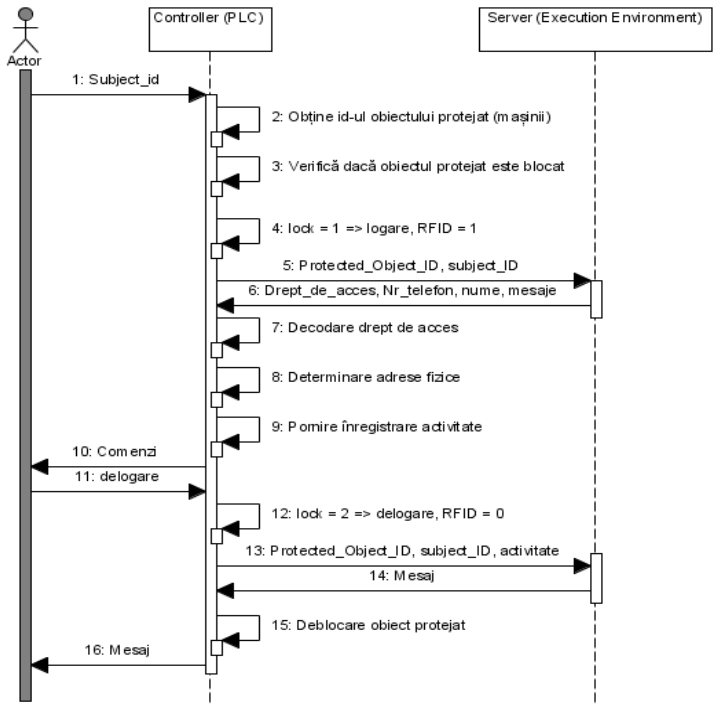


Fig. 3.14 Diagrama de secvențe simplificată pentru PLC

Așadar la nivelul PLC se dispune de trei ***sesiuni,*** Fig. 3.15:

- **Sesiune de logare** – care ajunge până la server. Prin intermediul acesteia se va determina dreptul unui subiect la respetivul obiect protejat. După decodarea dreptului subiectului și reobținerea adreselor fizice a intrărilor la care un subiect are drept de access urmează sesiunea de lucru.

- **Sesiune de lucru** – această sesiune este locală doar la nivelul PLC. În cadrul acestei sesiuni subiectul va putea accesa doar acele intrări pentru care are autorizarea corespunzătoare.

- **Sesiune de delogare** – ajunge până la server. În cadrul acesteia se vor realiza anumite operații de genul: deblocare subiect, deblocare obiect protejat, înscriere activitate subiect în baza de date.

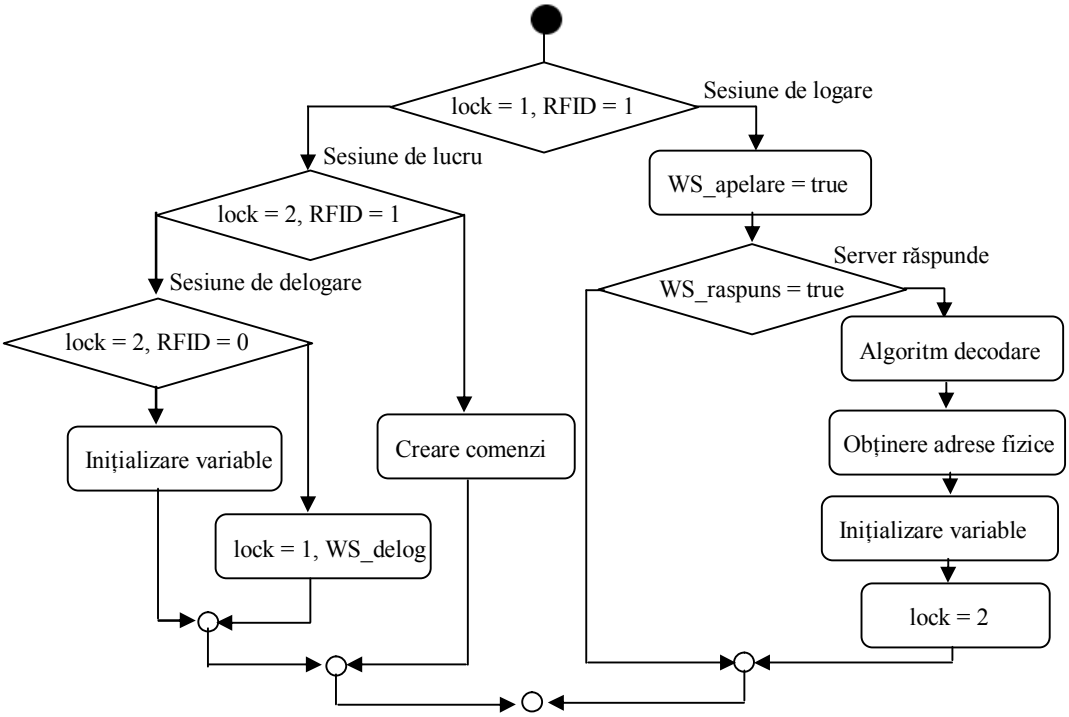


Fig. 3.15 Sesiuni la nivelul PLC

3.2.3 Algoritm codare - decodare drept de acces

Codarea într-un întreg a intrărilor la care un subiect are drept de acces prezintă avantajul că în acest fel nu trebuie să se vehiculeze între server și PLC adresele fizice ale intrărilor respective.

Un obiect protejat poate avea un număr foarte mare de intrări, iar vehicularea acestora devine din ce în ce mai greoaie cu cât acest număr crește. Totodată această codare ajută și în cazul în care comunicarea dintre server și PLC se face cu ajutorul unui Web Service, neâncărcând rețeaua de comunicare.

Pentru partea de codare a drepturilor pe care un subict le are s-a folosit o ierarhizare a adreselor fizice ale intrărilor obiectelor protejate, urmată de o transformare binar – integer [43]. De exemplu dacă obiectul protejat i dispune de j

intrări: $I_1 \ldots I_j$ se va considera că I_1 reprezintă bitul cel mai puțin semnificativ și I_j reprezintă bitul cel mai semnificativ.

Astfel formula de calcul al dreptului (right - R) dupa înlăturarea redundanțelor și a calificativelor expirate va fi (3.8) :

$$R = I_1 2^0 + I_2 2^1 + I_3 2^2 + \ldots + I_j 2^{j-1} = \sum_{i=1}^{j} I_i 2^{i-1} \quad (3.8)$$

Biții corespunzători intrărilor pentru care un subiect are access vor fi 1 iar restul biților vor fi zero.

Astfel, în cadrul unei sesiuni de logare se va obține dreptul unui subiect codat într-un întreg. La nivel PLC, pentru a reobține adresele fizice va trebui mai întâi să se decodeze acest drept, Fig. 3.16.

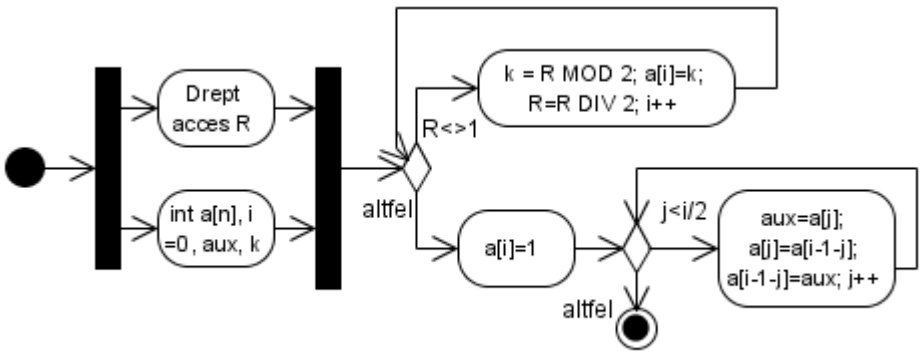


Fig. 3.16 Algoritm de decodare

În acest mod, în vectorul a[j] se reobține șirul de biți corespunzător (1 – are drept de acces 0 – nu are drept de acces). După decodare se pot determina adresele fizice ale intrărilor la care un subiect are drept de acces, precum și adresele ieșirilor ce vor fi comandate în cazul în care angajatul a apasăt una dintre intrarile permise.

De exemplu, pentru un PLC din familia Siemens, intrările și ieșirile sunt împărțite în grupuri de 8 intrări sau ieșiri:

- Ix.y unde I reprezinta tipul de adresă de tip input, x reprezintă byte-ul de adresă, iar y bit-ul de adresă
- Qz.k unde Q reprezinta tipul de adresă output, z byte de adresă, k bit de adresă

Dacă se consideră un exemplu simplu: pentru o mașină x sunt stocate într-un vector adresele inrărilor de care aceasta dispune (3.9)

Imachinex[10] = {I4.0, I0.1, I6.2, I0.3, I0.4, I0.5, I0.6, I0.7, I1.0, I1.1} (3.9)

Pentru aceeași mașină x sunt stocate într-un vector adresele ieșirilor care se comandă când respectiva intrare este activă (3.10):

Qmachinex[10] = {Q1.0, Q2.1, Q3.2, Q1.3, Q1.4, Q1.5, Q1.6, Q1.7, Q2.0, Q2.1 } (3.10)

Dacă dreptul de acces al subiectului yy primit de la server este R = 14, după decodarea acestuia se obține vectorul (3.11):

a[4] = {0, 1, 1, 1}, în acest caz n = 4 (3.11)

Se va folosi un algoritm de obținere al adreselor fizice a intrărilor la care angajatul are acces și al adreselor fizice al ieșirilor aferente ce vor fi comandate, Fig. 3.17.

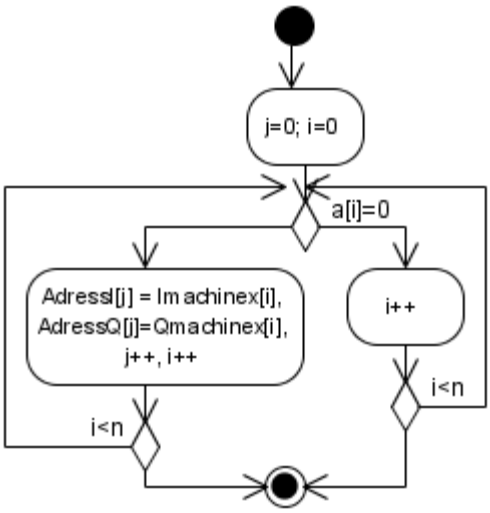


Fig. 3.17 Obținere adrese fizice

Astfel în vectorul AdressI[j] se vor obține adresele intrărilor la care angajatul are drept de acces corespunzător calificativelor lui.

In vectorul AdressQ[j] se vor obține adresele ieșirilor care pot fi comandate când intrarea aferentă este 1, respectiv un buton este apăsat.

În cazul exemplului considerat se va obține pentru mașina x, cu dreptul de access R = 14 următoarele valori (3.12):

AdressI[3] = { I0.1, I6.2, I0.3}
AdressQ[3] = {Q2.1, Q3.2, Q1.3 } (3.12)

Astfel când:
I0.1 = 1 -> Q2.1
I6.2 = 1 -> Q3.2

I0.3 = 1 -> Q1.3, restul intrărilor obiectului protejat respectiv vor fi fără autorizare pentru subiectul yy.

În acest mod se poate decide cine are drept de acces și care este dreptul subiectului respectiv. La nivel PLC nu se va avea cunoștință despre dreptul unui subiect, aceste drepturi și întregul proces aferent fiind stocate la nivel de server. La nivel PLC va trebui însă să existe aceeași ordonare a intrărilor unui obiect protejat pe care o avem și în partea de server, altfel întrega logică nu va mai funcționa corespunzător. Mai multe detalii legate de posibilități de codare, aplicații practice și diferite standarde pot fi găsite la [44].

În Fig. 3.18 se prezintă principalele operații necesare conceptului de autorizare bazat pe calificative și conectare fizică prin intermediul RFID, a cărui fază de dezvoltare a fost prezentată și va fi implementă în continuare.

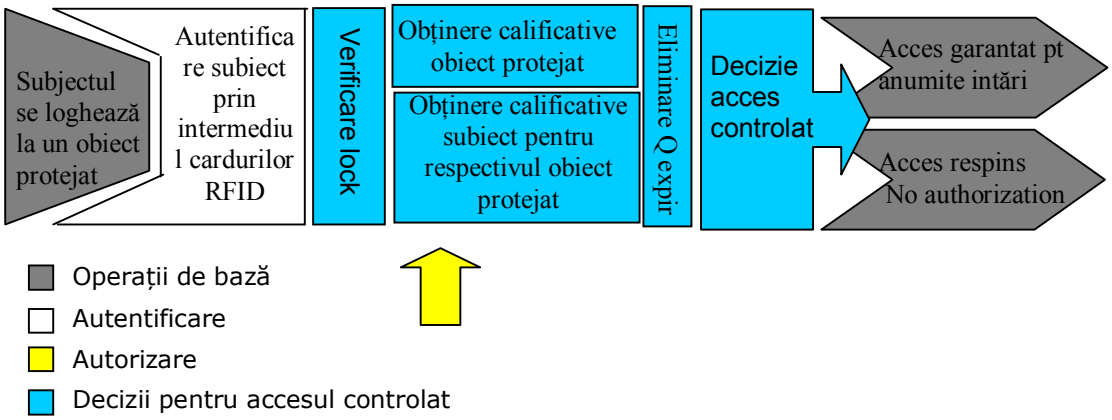


Fig. 3.18 Operații principale ale metodei de acces controlat

3.3 Sistem distribuit pentru implementarea QBAC, modelare matematică

În cadrul acestui subcapitol se va prezenta sistemul distribuit necesar implementării QBAC, structura simplificată sub formă de schemă bloc rezultată și modelul matematic aferent. Motivul realizării schemei bloc simplificate este dorința de a obține un model matematic ce descrie procesele cele mai importante din punct de vedere al metodei de acces la resurse dezvoltate.

3.3.1 Sistem distribuit pentru implementarea QBAC

Conform [45] definiția unui sistem distribuit este următoarea:

*„Prin **sistem distribuit** înțelegem un sistem implementat pe o rețea de calculatoare, în care componentele soft și hard, situate în calculatoarele din rețea*

comunică și își coordonează acțiunile numai prin intermediul transmiterii unor mesaje"

Exemple de sisteme distribuite se pot întâlni în multe aplicații, de la rețele bancare și până la sisteme industriale de comandă și control.

Caracteristicile principale ale unui sistem distribuit sunt [45], [46], [47], [48], [49]:

- **partajarea resurselor**, motivația principală a construirii și folosirii sistemelor distribuite;
- **inexistența unui ceas global**, acest lucru duce la folosirea mesajelor pentru cooperarea programelor;
- **eșecuri independente**, toate componentele sistemului pot eșua în funcționare.

În Fig. 3.19 se prezintă structura sistemului distribuit necesar pentru implementarea QBAC (respectiv QBAC extins).

Funcționalitățile diverselor subsisteme din cadrul structurii prezentate în Fig. 3.19 sunt următoarele:

1. La nivel de server sunt create toate obiectele de dezvoltare necesare (ex. clase, metode, baza de date, obiecte portal, Web Service, aplicație de administrare). Pentru implementarea acestuia se poate folosi orice limbaj de programare. Datorită faptului că acesta necesită o mare varietate de unelte de programare, inclusiv resurse umane, este recomandată folosirea unei platforme de integrare ERP, care ofera suport pentru programare orientată obiect, interfață web folosind MVC, suport pentru resurse umane, suport pentru creare calificativelor și a întregului proces aferent, suport SOA și portal. În acest mod se ușurează munca necesară implementării [51].

2. Stațiile de lucru din această arie vor fi folosite pentru programarea tuturor obiectelor de dezvoltare necesare, vor fi folosite de către administratori, de către cei care lucrează în birouri, etc.

3. La nivelul controller-elor se pot conecta cele n mașini ale căror intrări pot sa le acceseze angajații în funcție de calificativele de care aceștia dispun. Fiecare mașină are propriul cititor de carduri RFID, permițând astfel angajaților logarea și delogarea.

4. La acest nivel se face legătura dintre server și PLCs, dispunând totodată de posibilitatea de supraveghere și control a diferitelor date de proces.

5. Angajații și clienții pot accesa serverul prin intermediul internet-ului. Astfel, un client se poate informa asupra ofertelor firmei, în timp ce un administrator, de exemplu, se poate loga prin intermediul conexiunii VPN pentru a îndeplini anumite sarcini. De asemenea, folosind varianta de oferire

a drepturilor de acces și prin intermediul unui Web Service, se oferă suport și pentru obiectele protejate ce nu sunt neapărat localizate în cadrul firmei.

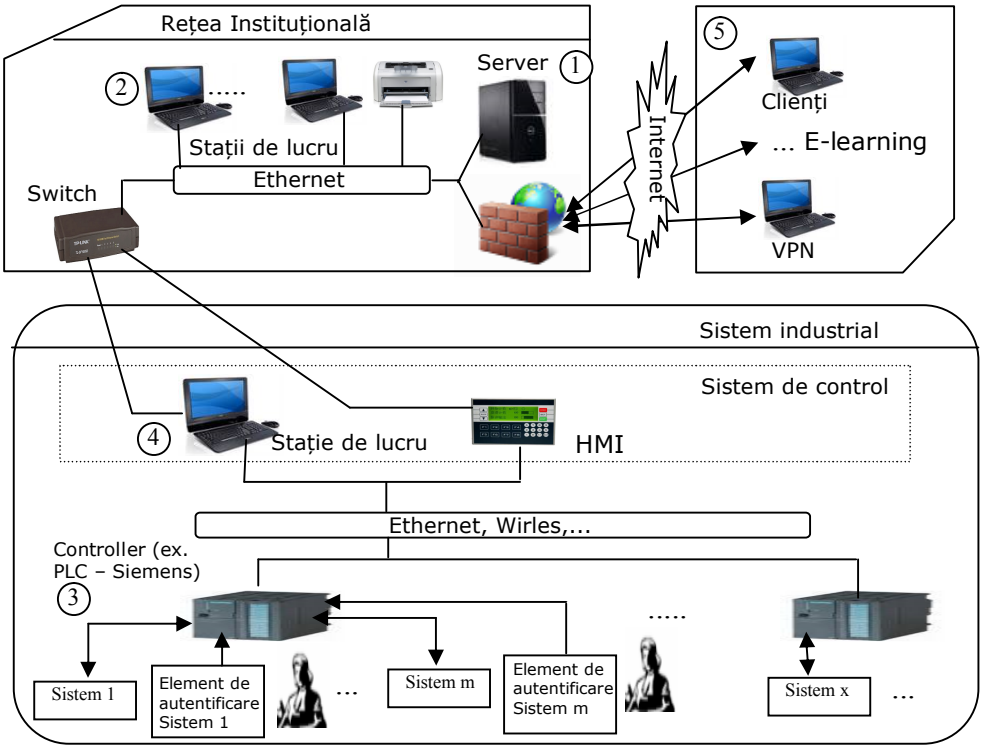


Fig. 3.19 Sistem distribuit pentru implementarea QBAC [50]

3.2.2 Structura sistemului distribuit sub formă de schemă bloc simplificată

În Fig. 3.20 se prezintă structura simplificată a sistemului distribuit necesar implementării QBAC.

S-a redus întregul sistem la elementele componente necesare la nivel de PLC, considerând ca și server platforma SAP NetWeaver, iar ca și client stația de lucru de la nivelul de control al procesului industrial.

După cum s-a precizat la nivelul PLC putem avea trei sesiuni: sesiune de logare, sesiune de delogare (care ajung până la server) și sesiune de lucru (locală, doar la nivelul PLC).

Din punct de vedere al **sesiunii de logare** și de **delogare** dispunem de un **sitem distribuit** de genul client – server, unde în mod principal server-ul este platforma SAP NetWeaver, iar clientul este stația de lucru. PLC-ul poate fi considerat la rândul lui tot un client deoarece acesta cere informații, iar stația de lucru devine în acest caz un server pentru PLC, deoarece aceasta oferă PLC-ului informații.

Prin urmare același proces poate fi considerat un server sau un client, în funcție de punctul de vedere din care se urmărește acel proces, adică al relației cauză -> efect.

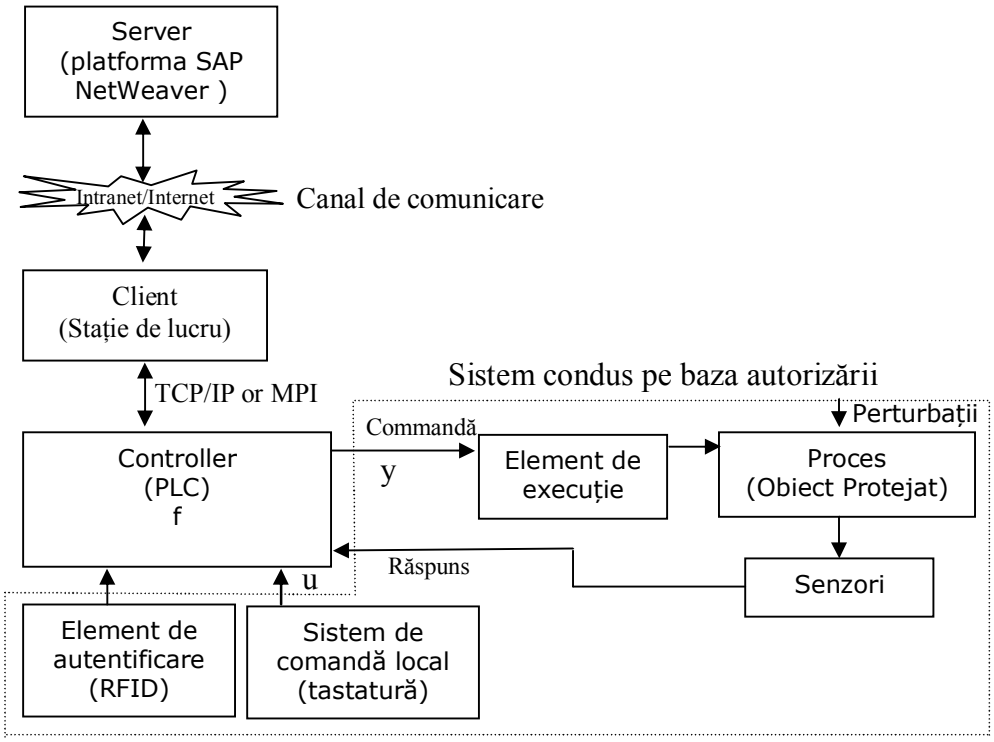


Fig. 3.20 Structura simplificată a sistemului distribuit

Un ***server*** îndeplinește următoarele funcții principale:

- primește o cerere de la un client;
- execută o operație pentru a putea răspunde clientului;
- trimite răspunsul clientului.

Un ***client*** îndeplinește următoarele funcții principiale:

- trimite mesaj la server;
- primește răspuns de la server.

În cazul de față, comunicarea dintre un client și un server poate fi descrisă ca și în Fig. 3.21 [52], [53].

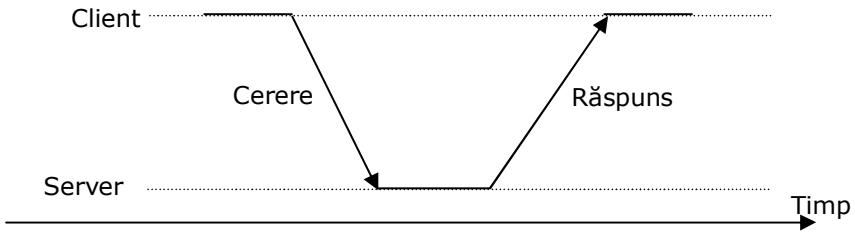


Fig. 3.21 Prezentare schematică a interacțiinii client - server

Dacă se privește sistemul din punct de vedere **al sesiunii de lucru** ce se desfășoară local, doar la nivel PLC se dispune de un **sistem** unde u reprezintă date de intrare, y reprezintă date de ieșire, iar f reprezintă funcția de prelucrare a datelor de intrare (3.13).

u = intrările accesate de către subiect
y=f(u)=bool(0,1), 1 drept de acces, 0 fără drept de acces (3.13)

Acest sistem ia în considerare și eventualele feed-back-uri de la senzori, precum și eventualele perturbații ce pot apărea. Dar, având în vedere că în metoda de acces controlat propusă, atât feed-back-urile cât și perturbațiile nu au importanță, nefăcând parte din logica QBAC, ele vor fi ignorate.

3.3.3 Modelul matematic rezultat

Nu întotdeauna se impune modelarea matematică a întregului sistem [54], [55], [56], în multe aplicații, printre care și cea dezvoltată în această teza, fiind suficientă modelarea unor procese care prezintă interes. Astfel vom privi sistemul nostru distribuit din punct de vedere al necesităților funcționării QBAC, respectiv cele trei sesiuni cu ajutorul cărora se acordă angajaților dreptul de a accesa doar acele resurse pentru care au calificativele (aptitudinile) corespunzătoare.

După cum s-a precizat, pentru sesiunile de logare și delogare se dispune de un sistem distribuit client - server ce are topologia de cominicare punct la punct.

Reprezentarea sistemului distribuit rezultat, se poate realiza prin intermediul grafului orientat G ce conține 3 noduri (sau vârfuri) și 4 arce, prezentat în Fig. 3.22.

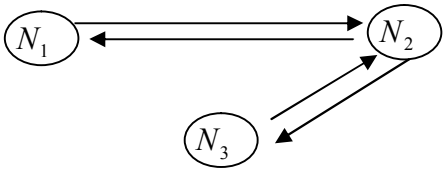


Fig. 3.22 Reprezentarea sub formă de graf a sistemului distribuit

unde:

N_1 – PLC

N_2 – Stație de lucru

N_3 – Server, ex. platforma SAP NetWeaver

Considerând graful orientat $G=(N,U),$ un arc al grafului G este $u \in U$, $u=(N_i,N_j)$. Un drum d al grafului G, reprezintă o succesiune de arce $(u_1,u_2,..u_j), j>1$ [45] iar lungimea drumului $l(d)$ reprezintă numărul arcelor pe care le conține drumul d. Pentru cazul grafului orientat G, reprezentarea arcelor, a drumului și a lungimii acestuia sunt prezentate prin intermediul relațiilor 3.14:

$$\begin{aligned} &N=\{N_1,N_2,N_3\},\\ &U=\{(N_1,N_2),(N_2,N_3),(N_3,N_2),(N_2,N_1)\}=\{u_1,u_2,u_3,u_4\}\\ &u_1=(N_1,N_2),u_2=(N_2,N_3),u_3=(N_3,N_2),u_4=(N_2,N_1)\\ &d=(u_1,u_2,u_3,u_4)\\ &l(d)=4 \end{aligned} \qquad (3.14)$$

unde:

u_i - arce

d - drum

$l(d)$ - lungimea drumului d

N : $|N|<\infty, N\neq \emptyset$, Mulțimea nodurilor sau vârfurilor

Pentru un arc $u_1=(N_1,N_2),$ N_1 este considerat extremitatea inițială a acestuia iar N_2 este considerat extremitatea finală a acestuia.

În ceea ce privește calculul drumului celui mai scurt între două noduri, în cazul de față nu apare nici o problemă, acesta fiind bine definit.

Matricea de adiacență [45], $A=(a_{k,m})$ ce definește graful G, este prezentată în tabelul de adevăr 3.15. Adiacența reprezintă proprietatea a două noduri de a fi unite printr-un arc.

$$A=(a_{k,m})= \begin{array}{c|ccc} & N_1 & N_2 & N_3 \\ \hline N_1 & 0 & 1 & 0 \\ N_2 & 1 & 0 & 1 \\ N_3 & 0 & 1 & 0 \end{array} \qquad (3.15)$$

unde:

$$a_{k,m} = \begin{cases} 1, dacă (N_k, N_m) \in U \\ 0, dacă (N_k, N_m) \notin U \end{cases} ; k, m = \overline{1,n}$$

Un sistem distribuit constă dintr-un set de procese secvențiale conectate într-o rețea iar comunicarea dintre acestea se realizează cu ajutorul mesajelor [45], [49]. Executarea unei aplicații folosind un astfel de sistem are ca și urmare producerea de evenimente: eveniment intern, eveniment de trimitere a unui mesaj și eveniment de recepționare a unui mesaj. Pentru a înțelege cât mai bine aceste evenimente, precum și succesiunea acestora, se realizează de cele mai multe ori diagrame de timp [45], [49], unde timpul este reprezentat pe orizontală, iar mesajele de comunicare între diferitele procese sunt reprezentate prin intermediul săgeților.

În Fig. 3.23 se prezintă diagrama de timp și relațiile dintre evenimente necesare comunicării dintre cele trei procese secvențiale ale sistemului distribuit rezultat.

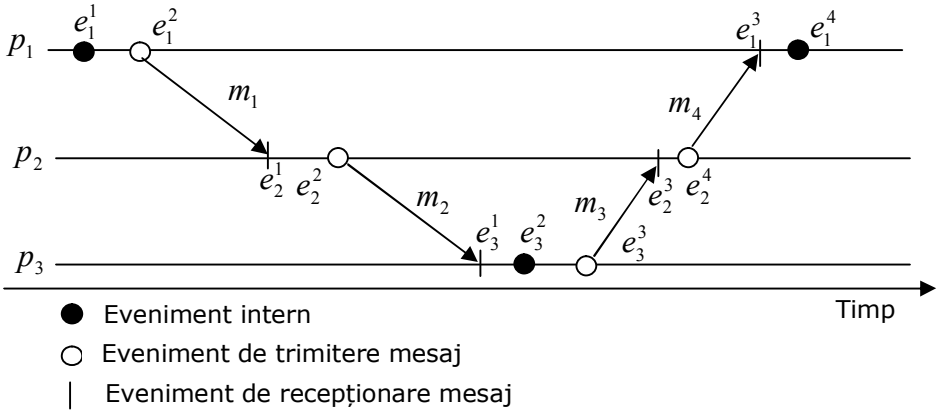


Fig. 3.23 Diagrama de timp, relații între evenimente

unde:

p_j - proces

$E = \{e_1^1, e_1^2, ..e_i^j\}$, mulțimea evenimentelor

m_i - mesaj transmis sau recepționat,

evenimentul - emisia mesajului m_i are ca urmare evenimentul – recepționare mesajului m_i de către un alt proces

Din punct de vedere al relațiilor dintre evenimente, dacă se introduce relația de cauzalitate $\rightarrow$ se poate constata (3.17):

$$e_1^1 \rightarrow e_3^2, e_3^2 \rightarrow e_1^4 \qquad (3.17)$$

Evenimentul e_1^1 poate să fie o sesiune de logare sau o sesiune de delogare. În cazul în care acest eveniment este o sesiune de logare, în urma producerii evenimentului e_1^4 se vor obține niște mărimi X , ce pot fi interpretate ca mărimi de stare pentru sistemul corespunzător sesiunii de lucru local de la nivelul PLC. Aceste mărimi vor defini vectorul de stare $x \in X$ pentru sistemul corespunzător sesiunii locale de la nivelul PLC-ului.

În ceea ce privește **sesiunea de lucru locală**, la nivelul PLC, avem un sistem proporțional, care se reduce în cazul QBAC la structura din Fig. 3.24.

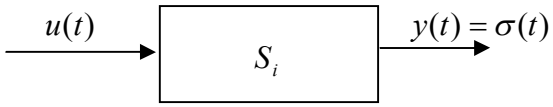


Fig. 3.24 Structura sistemului redus la sesiune de lucru locală la nivel PLC

Așadar, angajatul poate accesa intrările unui sistem S_i iar, dacă acesta are drept de acces vom trimite la ieșire 1, iar dacă nu are drept de acces vom trimite 0. Dreptul de acces a fost determinat la nivelul serverului pe baza calificativelor, iar adresele fizice ale intrărilor pentru care există drept de acces se găsesc în vectorul de stare $x \in X$. În cadrul sesiunii de lucru va avea loc doar procesul de verificare dacă intrarea accesată de angajat duce la trimiterea unei comenzi sau nu.

Vectorii de intrare și ieșire sunt prezentați în relațile 3.18 și 3.19.

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_i \end{bmatrix} = \begin{bmatrix} I_{x.1} \\ I_{x.2} \\ \dots \\ I_{x.j} \end{bmatrix}, \qquad (3.18)$$

$I_{x.j} \in (0,1)$ reprezintă adresa fizică a intrării sistemului S_i accesate de angajat

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_l \end{bmatrix} = \begin{bmatrix} Q_{x.0} \\ Q_{x.1} \\ .... \\ Q_{x.k} \end{bmatrix}, \qquad (3.19)$$

$Q_{x.k} \in (0,1)$, reprezintă adresa fizică a ieșirii sistemului S_i ce se comandă

Se pot considera semnalele de intrare și de ieșire ca și semnale standard treaptă unitară [54], acestea putând avea doar valori de 1 sau 0 (3.20).

$$\sigma(t)=\begin{cases}0, t<0\\ 1, t\geq 0\end{cases} \quad (3.20)$$

Mărimile de stare ale sistemului sunt cele care caracterizează starea sistemului la un moment dat și care, împreuna cu mărimea de intrare u ne permit la orice moment $t \in T$ să determinăm mărimea de ieșire a sistemului. Mărimile de stare ale sistemului S_i reprezintă intrările pentru care angajatul XX are drept de access, mărimi obținute în urma unei sesiuni de logare (3.21).

$$X=\begin{bmatrix}x_1\\ x_2\\ \dots\\ x_n\end{bmatrix}=\begin{bmatrix}I_{n..0}\\ I_{n.1}\\ \dots\\ I_{n.x}\end{bmatrix},\ I_{n.x} \text{ adresa fizică a intrării cu drept de acces} \quad (3.21)$$

Relațiile 3.22 definesc ieșirea sistemului în funcție de intrarea accesată și de mărimile de stare.

$$y=\begin{cases}1, u_i=1 AND I_{x.j}=I_{n.x}\\ 0, \begin{cases}u_i=1 AND I_{x.j}\neq I_{n.x}\\ u_i=0\end{cases}\end{cases} \quad (3.22)$$

Ca exemplu se consideră patru intrari u1 – u4 ale sistemului S_i, ce au adresele fizice de forma binară prezentate în tabelul 3.2.

B_0	B_1	**u**
0	0	u_1
0	1	u_2
1	0	u_3
1	1	u_4

Tabelul 3.2 Definirea a patru intrări ipotetice ale sistemului S_i

În cele ce urmează vom presupune că la un anumit moment de timp t_0 poate să apară doar o singură cerere de acces a unui angajat pentru sistemul S_i. În aceste condiții vom presupune ca avem tabelul de adevăr 3.3.

B_0	B_1	**u**
0	0	$u_1=1$
0	1	$u_2=0$
1	0	$u_3=0$
1	1	$u_4=0$

Tabelul 3.3 Tabel de adevăr pentru intrări

În aceste condiții, se pote defini funcția logică sub forma disjunctivă prezentată în relația 3.23:

$$u = \overline{B0} \cdot \overline{B1} \tag{3.23}$$

Pe de alta parte, se va considera că vectorul de stare are și el patru stări definite pe baza tabelului 3.4.

A_0	A_1	**x**
0	0	x_2
0	1	x_1
1	0	x_3
1	1	x_4

Tabelul 3.4 Definirea a patru stări ipotetice pentru sistemul S_i

În aceste condiții, dacă $B_0 = A_0$ și $B_1 = A_1$ atunci $x_2 = 1$ ceea ce implică relația 3.24.

$$x = \overline{A0} \cdot \overline{A1} \tag{3.24}$$

În mod identic, presupunem că ieșirea este descrisă de tabelul de adevăr 3.5.

u	**x**	**y**
0	0	$y_4 = 0$
0	1	$y_1 = 0$
1	0	$y_2 = 0$
1	1	$y_3 = 1$

Tabelul 3.5 Tabelul de adevăr pentru ieșiri

Pe baza celor menționate, funcția logică a ieșirii este prezentată în relația 3.25.

$$y = y_3 = u \cdot x \qquad (3.25)$$

Așadar, din punct de vedere matematic, sistemul S_i poate fi reprezentat de următorul sextuplu:

$$S_i = \{T, U, U^1, X, Y, Y^1\}$$

unde:
$T \subset R$ - mulțimea de timp,
U – mulțimea valorilor variabilei de intrare $u \in U$,
U^1 - clasa funcțiilor de intrare admise de sistem – doar semnal treaptă unitară
X – spațiul stărilor, $x \in X$
Y – mulțimea valorilor variabilei de ieșire $y \in Y$
$Y^1 = \{y(t) : T \to Y\}$ - clasa funcțiilor de ieșire admise de sistem – doar semnal treaptă unitară

Concluzii:

În cadrul acestui capitol s-a:

- prezentat (prin intermediul diagramelor de clase UML și folosind un exemplu din lumea reală) structura pattern-ului QBAC creat. Totodată s-au scos în evidență avantajele acestuia și modul în care poate fi combinat cu pattern-ul RBAC în vederea obținerii soluțiilor moderne de învățare, pentru administrare, funcționalități de genul ESS;

- s-a prezentat prin intermediul Data Mining și diagrame UML faza de dezvoltare al pattern-ului QBAC, astfel încât acesta să poată fi implementat folosind orice limbaj de programare. S-a scos totodată în evidență necesitatea folosirii unei platforme ERP, aceasta accelerând procesul implementării și oferind soluții complexe;

- s-a propus structura sistemului distribuit necesar implementării QBAC;

- s-a realizat modelarea matematică.

Concluzionând, se poate spune că pattern-ul QBAC realizat îmbină două problematici actuale: accesul controlat la resurse și procesul continuu de învățare al angajaților, aducând totodată obiectele protejate în sfera resurselor umane. Astfel, prin intermediul calificativelor se poate răspunde la întrebări de genul: „Ce calificative îi lipsesc subiectului S pentru a obține Job-ul Y ?”

Prin intermediul algoritmului de codare propus, pentru comunicarea datelor între server (Platforma SAP NetWeaver) și PLC s-a realizat o codare binar – integer.

În acest mod se evită vehicularea adreselor fizice ale intrărilor pentru care un subiect are drept de acces. Acestă codare este avantajoasă deoarece nu se încarcă rețeaua de comunicare, ajutând totodată și în procesul de integrare a noi obiecte protejate și intrări ale obiectelor protejate deoarece, prin realizarea codării nu trebuiesc făcute modificări în variabilele globale ale PLC-ului. Dacă aceste modificări ar trebui făcute de fiecare dată când apare o modificare, s-ar consuma foarte mult efort din partea unui programator, acesta trebuind în acest caz să modifice variabilele globale PLC și aliace-urile din OPC. Așadar, prin intermediul algoritmului de codare propus, se minimizează pe cât posibil efortul necesar implementării schimbărilor aduse la nivel de obiect protejat și nu se încarcă rețeaua de comunicare dintre PLC și server.

4. PLATFORMA SAP NETWEAVER

În cele ce urmează se va prezenta succint platforma de integrare și de aplicații SAP NetWeaver, scoțând în evidență motivele pentru care a fost alesă în vederea implementării metodei de acces controlat. Apoi, se va analiza modul în care această platformă ne ajută în implementarea QBAC, făcându-se totodată și o analiză în vederea alegerii componentelor și uneltelor SAP ce pot fi folosite în vederea accelerării implementării și a obținerii rezultatelor optimale.

Marea parte a acestui capitol se va dedica apoi prezentării modului de implementare în cadrul acestei platforme al diverselor obiecte de dezvoltare necesare la nivelul serverului, precum și modul în care s-au internaționalizat diversele obiecte realizate.

4.1 Justificarea alegerii platformei SAP NetWeaver

SAP NetWeaver [57] este platforma de integrare și de aplicații SAP, bază pentru SAP Business Suite [58], ce oferă companiilor soluții viabile pentru controlarea proceselor economice în vederea creșterii productivității. Această platformă dispune de patru nivele (layers): ***integrarea persoanelor*** (People Integration), ***integrarea informației*** (Information Integration), ***integrarea proceselor*** (Process Integration) și ***platforma de aplicații*** (Application Platform), Fig. 4.1 (adaptată din [59]).

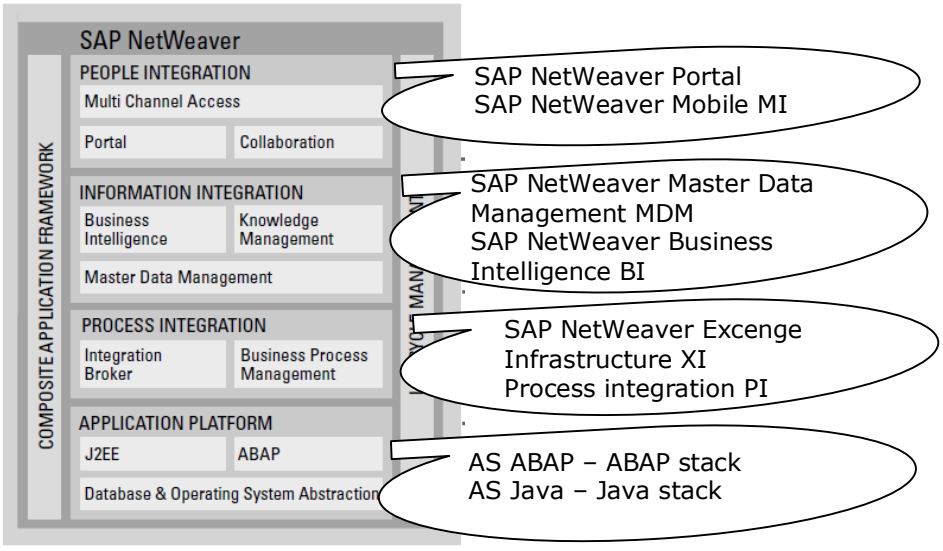


Fig. 4.1 Platforma SAP NetWeaver

După cum se poate observa, SAP NetWeaver are la bază o platformă de aplicații ce joacă un rol central avându-se în vedere că fără aceasta nu s-ar dispune de medii de programare, wizards, frameworks, etc. și nici funcționarea celorlate componente (ex. SAP NetWeaver Portal, SAP ERP HCM) nu ar fi posibilă. La rândul ei platforma de aplicații este formată din serverul de aplicații ABAP (AS ABAP) și serverul de aplicații Java (AS Java). AS ABAP este denumit și ABAP stack, oferind ABAP ca și limbaj de programare, iar AS Java este denumit și Java stack, oferind Java ca și limbaj de programare.

Pentru a se crea aplicații ABAP se folosește mediul de dezvoltare ABAP Workbench, iar pentru a se crea aplicații având ca și limbaj de programare Java se folosește SAP NetWeaver Developer Studio.

Câteva dintre ***motivele*** pentru care s-a ales platforma de integrare și aplicații SAP pentru implementare modelului de acces controlat QBAC propus sunt:

- SAP NetWeaver oferă o largă varietate de așa numite usages ce pot fi folosite astfel încât să se ușureze efortul necesar implementării. De exemplu, prin folosirea SAP NetWeaver Portal se poate crea portalul firmei, oferind totodată subiecților posibilitatea de a rezerva un curs, de a participa la E-learning, sau de a oferi administratorului nivelului celui mai de jos posibilitatea de a accesa aplicația de administrare în portal.

- Este bază pentru SAP Business Suite, de unde se va folosi SAP ERP HCM.

- Oferă posibilitatea de a lucra în mod natural cu calificative. Acestea sunt folosite în procesele de resurse umane, neavând nimic de-a face cu autorizarea. Faptul că nu trebuie să se realizeze propria aplicație prin care să se creeze obiecte de tip calificative va fi de un ajutor deosebit. Așadar, se va folosi componenta standard SAP pentru a genera calificativele, iar apoi se va crea întregul proces (ex. Bază de date, clase) necesar introducerii acestora în mecanismul de acces controlat al QBAC (care nu face parte din standardul SAP fiind contribuția autorului).

- Dispunem de o largă gamă de unelte și tehnici care ne ajută în procesul de programare. De exemplu suport MVC, ușurință în folosirea de Web Service-urilor, posibilitatea de lucru cu ActiveX, suport pentru OOP, posibilitate de creare a bazelor de date relaționale.

- Se dispune de subcomponenta SAP standard HR (ce face parte din SAP ERP HCM) pentru crearea datelor angajaților, astfel încât să nu fie necesar să se creeze propria aplicație necesară administratorului acestui nivel.

- Se dispune de unelte necesare pentru a crea diversele obiecte de dezvoltare multilinguale, astfel încât funcționalitățile create să "vorbească" limbile necesare celor cărora li se adresează.

După alegerea platformei de integrare a trebuit să se aleagă serverul de aplicații (AS ABAP sau AS Java) pentru implementarea funcționalităților de bază (ex. clase, web service, bază de date). Datorită faptului că avem nevoie de componenta SAP ERP HCM pentru a crea datele subiecților și calificativele, alegerea serverului de aplicații ABAP a fost inevitabilă.

SAP NetWeaver Developer Studio cu programare Java [60], oferă și ea tot ceea ce este nevoie pentru crearea logicii QBAC (ex. clase, bază de date). Acesta are la bază Eclipse și pune la dispoziție așa numitele perspective pentru îndeplinirea diferitelor task-uri. Exemple de ***perspective***:

- **Web Dynpro** [61], [62] oferă toate uneltele cu ajutorul cărora se pot crea aplicații web folosind principiul MVC, iar pentru ușurarea muncii de programator se dispune de o serie de unelte adiționale, ca de exemplu Data Modeler și View Designer.

- **Dictionary** pune la dispoziție uneltele necesare creării diverselor obiecte necesare bazelor de date, ca de exemplu obiecte de tip tabele, tipuri de dată simple și structuri, indecși.

Nu s-a ales însă să se lucreze cu Java deoarece SAP ERP HCM folosește AS ABAP pentru stocarea datelor, iar în cazul în care s-ar fi lucrat cu Java ar fi trebuit să se creeze o comunicare între AS ABAP și AS Java astfel încât să se obțină datele din cadrul HCM (cele două AS folosesc scheme diferite de baze de date). Astfel s-ar fi complicat lucrurile, toate acestea afectând negativ viteza de execuție.

Folosind AS ABAP, toate elementele de dezvoltare necesare pentru QBAC vor fi create în cadrul aceluiași server de aplicații și se va putea lucra nativ cu aceste date fără extra funcționalități de comunicare între AS ABAP și AS Java. De asemenea baza de date necesară creării modelului de acces controlat va putea fi foarte ușor legată de bazele de date generate de modulul standard SAP.

După cum se poate observa în Fig. 4.2 (adaptată din [59]) AS ABAP dispune de trei nivele: nivelul de **prezentare** (Presentation layer), nivelul **business** (Busienss Layer) și nivelul de **persistență (**Persistance Layer).

Așadar, cu ajutorul celor patru nivele puse la dispoziție de platforma SAP NetWeaver se pot realiza toate elementele de dezvoltare necesare creării logicii QBAC. Cu ajutorul SAP NetWeaver portal se poate realiza varianta extinsă QBAC și se poate integra aplicația de administrare a nivelului de interfață cu obiectele protejate. Prin folosirea modului ***SAP ERP HCM*** (ce are la bază platforma SAP NetWeaver) se vor putea crea și administra datele legate de subiecți, cursuri și calificative (ex. apartenență în cadrul companiei, date personale, calificare și recalificare, cursuri) precum și structura organizatorică a companiei.

În Fig. 4.3 se prezintă modulele și părțile din SAP NetWeaver folosite pentru scopul QBAC. În partea de integrare a persoanelor, Collaboration face parte tot din SAP NetWeaver Portal dar nu a fost folosit pentru scopul acestui proiect. De asemenea SAP ERP conține nu doar componenta HCM, iar HCM conține la rândul lui mai multe funcționalități, în acest caz prezentându-se doar ceea ce s-a folosit pentru scopul acestui proiect.

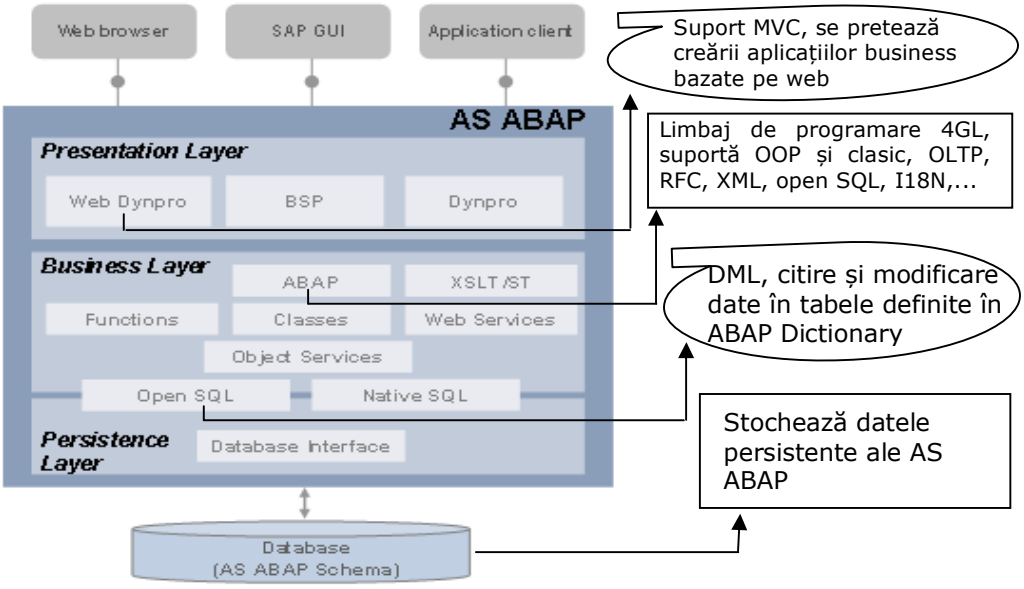


Fig. 4.2 Structura AS ABAP

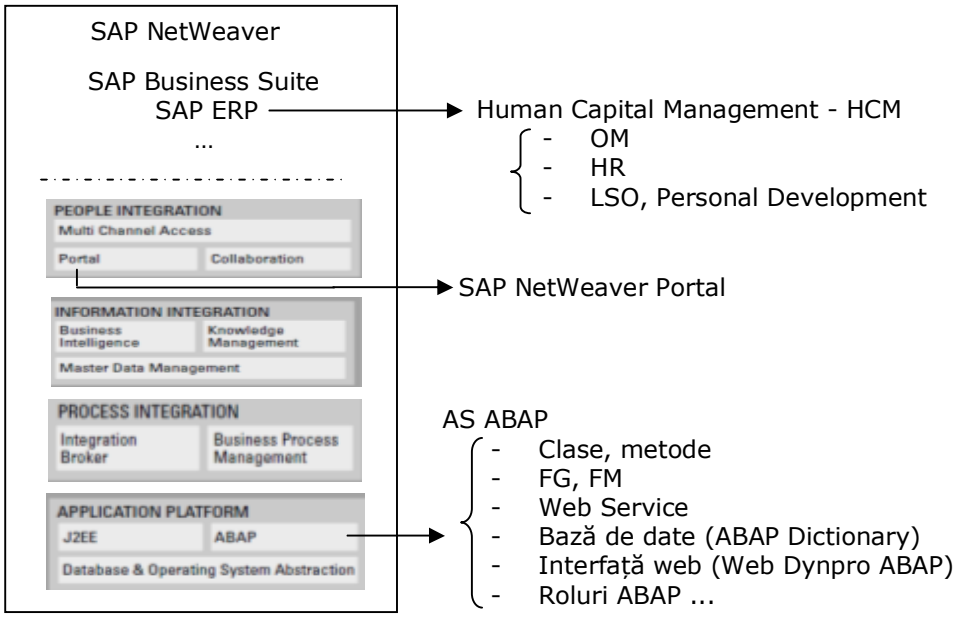


Fig. 4.3 Module și unelte utilizate pentru implementarea metodei de acces controlat bazată pe calificative

Serverul de aplicații Java va fi și el folosit pentru implementarea QBAC, dar în mod transparent, neavând nevoie de programare Java. De exemplu:

- componenta SAP NetWeaver Portal rulează pe stack-ul Java. Posibitățile de realizare a comunicării cu back-end ABAP este oferită însă în mod standard de către această componentă;
- când se testează un Web Service în cadrul ABAP Workbench se folosește o pagină de test (Web Services Navigator) ce rulează pe stack-ul Java;
- când se folosește ADS (pentru a crea formulare Adobe în Web Dynpro ABAP) se va folosi tot AS Java, deoarece acesta este instalată pe stack-ul Java.

4.2 Managementul Capitalului Uman (SAP ERP HCM)

Piața produselor ERP este dominată de patru mari dezvoltatori: SAP, Oracle, Microsoft și IBM [63].

SAP ERP HCM poate fi încadrată în categoria celor mai complexe componentele SAP ERP, dispunând de o largă gamă de funcții (componente) dintre care, pentru scopul acestui proiect s-a folosit:

- **Organization Management OM** [64], prin intermediul căreia se oferă posibiliatea creării datelor companiei.
- **Human Resources HR** [65], prin intermediul căreia se oferă posibilitatea creării datelor subiecților.
- **Learning Solutions LSO, Personal Development** [66], [67], prin intermediul cărora se oferă posibilitatea creării calificativelor și a întregului proces aferent.

În cadrul acestei componente ERP o importanță deosebită o au așa numitele ***infotypes.*** Toate datele care pot fi grupate din punct de vedere logic sunt integrate aici într-un infotype. Atunci când un infotype conține o largă gamă de informații, acesta poate fi subdivizat în așa numitele ***subtypes***.

Pentru a introduce date în infotypes se dispune de screen-uri prin intermediul cărora se introduc practic acele valori ca și înregistrări în baza de date aferentă. Așadar, fiecare infotype este definit ca tabelă în ABAP Dictionary și este identificat prin intermediul unui număr unic format din patru cifre. Fiecare infotype dispune de asemenea de o dată de început a valabilității acelor date și o dată de încheiere a valabilității. Această proprietate va fi foarte folositoare pentru logica QBAC în cazul calificativelor.

Acest concept (inotypes) este valabil pentru toate componenentele HCM, indiferent dacă este OM, HR, LSO sau altele. Toate acestea se îmbină practic ca un întreg, rezultând o mare bază de date interconectată: calificativele sunt legate de subiecți, subiecții (angajații) fac parte dintr-o anumită companie, etc.

Folosind componenta SAP ERP HCM se facilitează nu numai posibilitatea creării datele necesare fără efort de programare dar și crearea bazelor de date aferente. Totodată beneficiem și de o serie de ***avantaje*** ca de exemplu:

- un concept de autorizare adecvat care să protejeze datele create;
- capabilități avansate de căutare: Rapoarte (Repors) și Interogări (Querys) [68], [69];
- suport pentru particularitățile diferitelor țări;
- posibilitate de adaptare la necesitățile fiecărei companii prin intermediul customizing (standard), sau enhancement (ca și necesități specifice unui client). Pentru cazul nostru un simplu customizing sau enhancement nu este suficient.

În continuare se vor prezenta datele de test ale subiecților precum și întregul proces necesar calificativelor. Nu se va prezenta partea de OM deoarece aceasta reprezintă doar bază pentru logica QBAC, în sensul că subiecții trebuie să aparțină unei compani și dispun de unul sau mai multe job-uri, dar acestea nu intră în mod direct în logica metodei de acces controlat QBAC ci doar în mod pasiv.

Pentru a avea o imagine de ansamblu asupra straturilor (layers) platformei SAP NetWeaver, folosite de către componentele SAP ERP și în particular SAP ERP HCM se pot consulta harțile SAP ERP Solution Map și SAP ERP HCM Solution Map, care pot fi găsite și pe seit-ul SDN.

4.2.1 Crearea datelor de test folosind Human Resources (HR)

După cum s-a precizat deja, s-a folosit această componentă HCM pentru a crea toate datele subiecților [70]. În acest mod, pentru crearea și administrarea acestor date nu trebuie să se creeze propria aplicație, dispunând în acest caz de un sistem modern și complex pe care putem să îl adaptăm conform necesităților.

O importanță deosebită o are cheia unică a datelor fiecărui subiect, formată din opt cifre – PERNR (Personal Number). Acest număr reprezintă identitatea unui subiect în cadrul companiei.

Pentru scopul testării metodei de acces controlat QBAC s-au creat datele a 11 subiecți de test, prezentate în Fig. 4.4.

Așadar, se oferă o largă varietate de screen-uri prin intermediul cărora se pot introduce valori în infotypes. În exemplul din figura prezentată, se poate căuta după id-ul unui subiect și apoi se pot introduce valori în diferitete infotypes pentru respectivul subiect. În partea de jos a screen-ului prezentat se poate căuta după diferitele infotypes. În funcție de necesități și de componentele HCM folosite se pot avea infotypes de la 0000 până la 9999. De exemplu HR are rezervate infotypes 0000 – 0999.

După cum s-a precizat, datele introduse în infotypes sunt inserate apoi ca și înregistrări în tabelele corespunzătoare. Se va prezenta un singur exemplu, și anume structura tabelului PA002 (Infotype Personal Data) din ABAP Dictionary (Fig. 4.5).

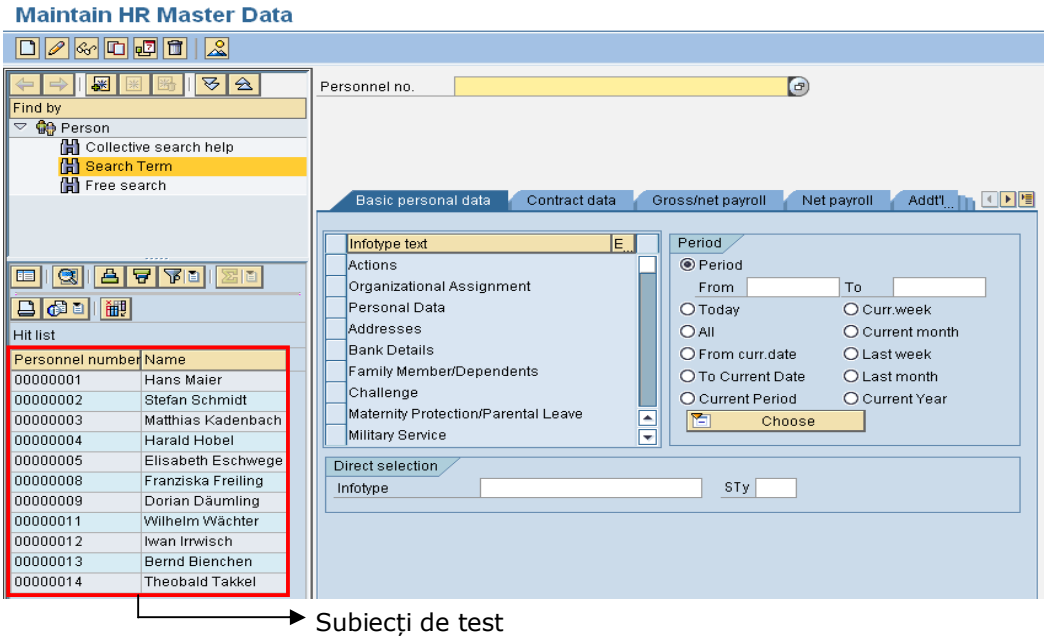


Fig. 4.4 Subiecți de test în HR

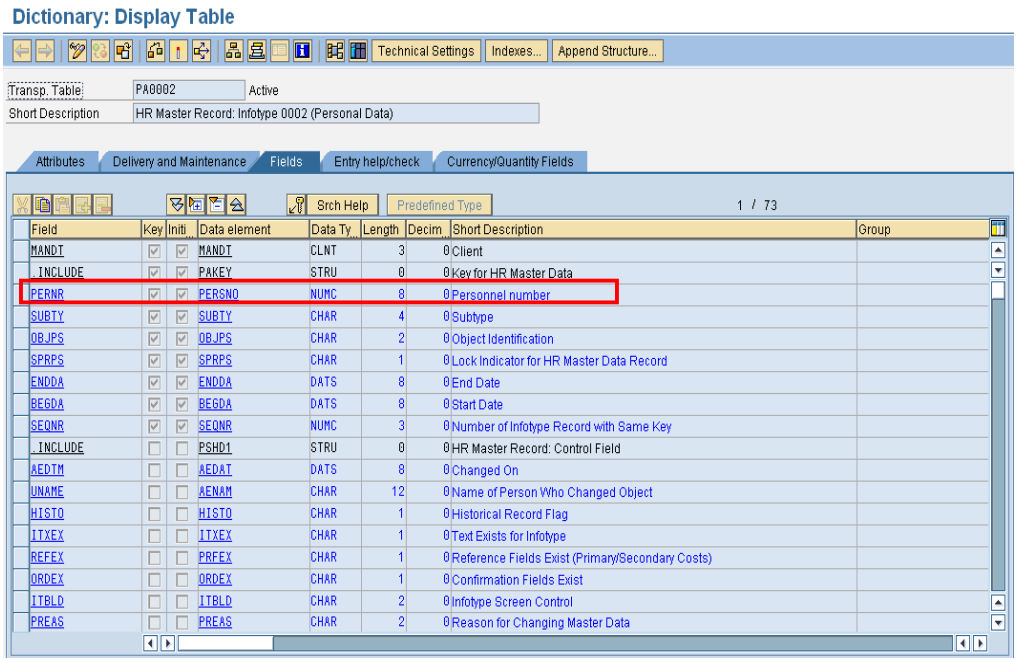


Fig. 4.5 Structura DDIC a tabelului corespunzător infotype 0002 (date personale)

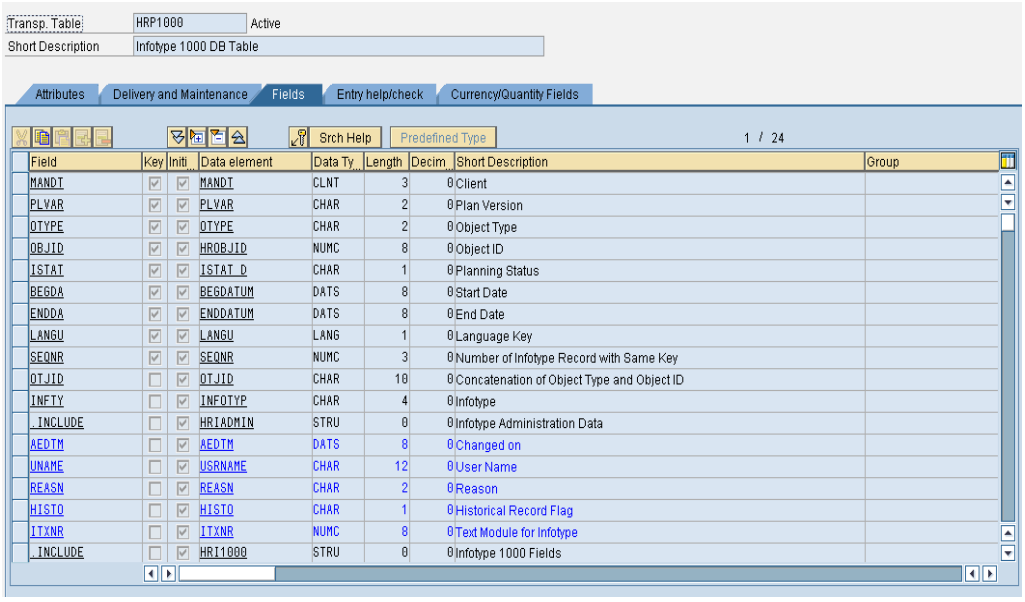
După cum se poate observa, PERNR reprezintă cheia unică. Această cheie are un rol primar în HCM și va fi întipărită și pe cardurile de identificare RFID cu care se vor autentifica subiecții la nivelul obiectelor protejate. Această cheie va reprezenta așadar și cheia de identificare a subiecților în Framework-ul QBAC. Un alt motiv pentru care s-a prezentat structura acestei tabele este acela că prin intermediul acesteia s-a realizat legătura cu datele necesare din HR.

Concluzionând, pentru realizarea logicii QBAC, atunci când este nevoie de date ce aparțin HCM se vor folosi datele stocate în baza de date aferentă (AS ABAP). Astfel, va trebui să se realizeze o legătură între baza de date necesară pentru integrarea obiectelor protejate și baza de date corespunzătoare HCM. Această legătură se va face prin intermediul interconectării tabelelor corespunzătoare.

4.2.2 Crearea calificativelor și a procesului aferent

Și în acest caz, folosind funcționalitățile SAP ERP HCM, nu va mai trebui creată aplicația prin intermediul căreia să se creeze calificativele și procesul aferent acestora. Așadar, și în acest caz se va beneficia de funcționalități complexe astfel încât să se obțină soluții competitive, soluții ce sunt folosite în sistemele productive reale.

Un rol deosebit îl are cheia fiecărui calificativ. În cadrul acestui modul un calificativ este un obiect de tip Q iar o grupă de calificative este un obiect de tip QK. În tabela HRP1000 (Fig. 4.6) se pot găsi toate tipurile de obiecte create în sistem împreună cu descrierea acestora.



Field	Key	Initi	Data element	Data Ty	Length	Decim	Short Description	Group
MANDT	☑	☑	MANDT	CLNT	3	0	Client	
PLVAR	☑	☑	PLVAR	CHAR	2	0	Plan Version	
OTYPE	☑	☑	OTYPE	CHAR	2	0	Object Type	
OBJID	☑	☑	HROBJID	NUMC	8	0	Object ID	
ISTAT	☑	☑	ISTAT_D	CHAR	1	0	Planning Status	
BEGDA	☑	☑	BEGDATUM	DATS	8	0	Start Date	
ENDDA	☑	☑	ENDDATUM	DATS	8	0	End Date	
LANGU	☑	☑	LANGU	LANG	1	0	Language Key	
SEQNR	☑	☑	SEQNR	NUMC	3	0	Number of Infotype Record with Same Key	
OTJID	☐	☑	OTJID	CHAR	10	0	Concatenation of Object Type and Object ID	
INFTY	☐	☑	INFOTYP	CHAR	4	0	Infotype	
.INCLUDE	☐	☑	HRIADMIN	STRU	0	0	Infotype Administration Data	
AEDTM	☐	☑	AEDTM	DATS	8	0	Changed on	
UNAME	☐	☑	USRNAME	CHAR	12	0	User Name	
REASN	☐	☑	REASN	CHAR	2	0	Reason	
HISTO	☐	☑	HISTO	CHAR	1	0	Historical Record Flag	
ITXNR	☐	☑	ITXNR	NUMC	8	0	Text Module for Infotype	
.INCLUDE	☐	☑	HRI1000	STRU	0	0	Infotype 1000 Fields	

Fig. 4.6 Structura tabelului HRP1000

Cele mai reprezentative informații pentru implementarea QBAC sunt: câmpul OTYPE unde este păstrat tipul obiectelor (ex. Q - Qualifications, QK – Qualification Group, ET – E-learning), câmpul OBJID, unde este păstrat id-ul obiectului (ex. 50000219, 50000230) și câmpurile BEGDA și respectiv ENDDA unde sunt păstrate datele de început și de sfârșit a valabilității obiectului cu cheia respectivă.

Cu ajutorul câmpurilor BEGDA și ENDDA se va putea acorda subiecților calificative care au proprietatea că pot expira după o perioadă de timp. Astfel, pentru a determina dreptul de acces al unui subiect pentru deservirea intrărilor unui obiect protejat se vor elimina calificativele care sunt expirate. Pentru ca un subiect să aibă timp sa își refacă școlarizarea, înainte cu 20 de zile ca un calificativ să expire se va informa respectivul subiect.

În continuare s-a creat catalogul de calificative pentru cinci mașini de test. Fizic se va dispune decât de o singură mașină de test. Motivul pentru care s-au creat mai multe mașini de test este de a verifica algoritmul, care este realizat general, indiferent de câte mașini de test se dispune.

După cum s-a prezentat în faza de dezvoltare, pentru o mai bună definire, mașinile sunt împărțite în grupe de mașini: grup_mașini101 și grup_mașini102. Pentru fiecare mașină s-au creat câte trei calificative: Installer, Operator și Tool_Setter, Fig. 4.7.

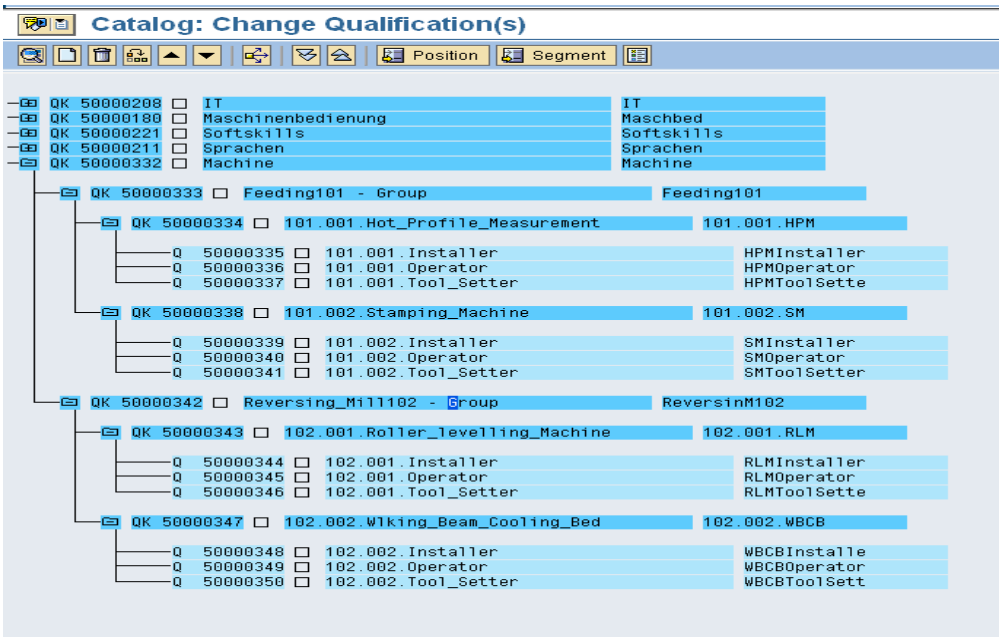


Fig. 4.7 Structura catalogului de calificative creat pentru mașinile de test

Pentru un calificativ trebuie să stabilim și o scală de apreciere ca de exemplu: criterii standard 1 - 6, criterii standard 6 – 1, scală standard 1-10, scală DA/Nu (admins/respins). Deoarece scala DA/NU se potrivește foarte bine cerințelor QBAC s-a ales ca toate calificativele create în catalogul de calificative pentru mașini să corespundă acestei scale.

Catalogul de calificative rezultat este așadar format din grupe de calificative și calificative, Fig. 4.8.

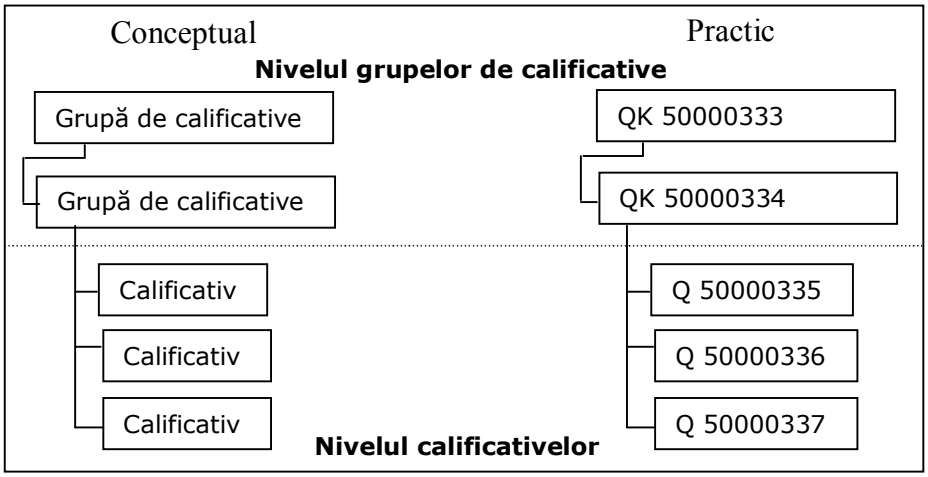


Fig. 4.8 Relație calificative - grupe de calificative

După crearea catalogului de calificative urmează să se creeze catalogul de cursuri corespunzător. Astfel, angajații pot să participe la școlarizare și, după îndeplinirea cerințelor ce se cer, să obțină calificativele necesare determinării dreptului de acces la intrările obiectelor protejate.

Pentru crearea catalogului de cursuri necesar realizarii școlarizării pentru angajați s-a folosit structura:

Mașini->Group_mașini->Mașina_nume->Curs_nume

În Fig. 4.9 se prezintă catalogul de cursuri corespunzător mașinilor de test. Pentru fiecare curs se pot crea mai multe tipuri de școlarizare ca de exemplu: școlarizare bazată pe Web, școlarizare în săli de clasă. Pentru implementarea QBAC s-a folosit școlarizarea în săli de clasă. Pentru QBAC extins cu E-learning trebuie să se realizeze școlarizare bazată pe web. După realizarea catalogului de cursuri se poate seta pentru fiecare curs o serie de proprietăți ca de exemplu: perioada calendaristică, când se va desfășura acel curs, costul respectivului curs (pentru angajații interni și externi), locația cursului, limba în care va fi ținut respectivul curs, capacitatea minimă, maximă și optimală pentru respectivul curs.

După ce s-au stabilit proprietățile cursurilor, angajații vor putea rezerva locuri la cursurile a căror calificative doresc să le obțină. De exemplu, în Fig. 4.10 se prezintă modul în care se rezervă un loc la cursul 101.101.Installer pentru angajatul cu id-ul 12.

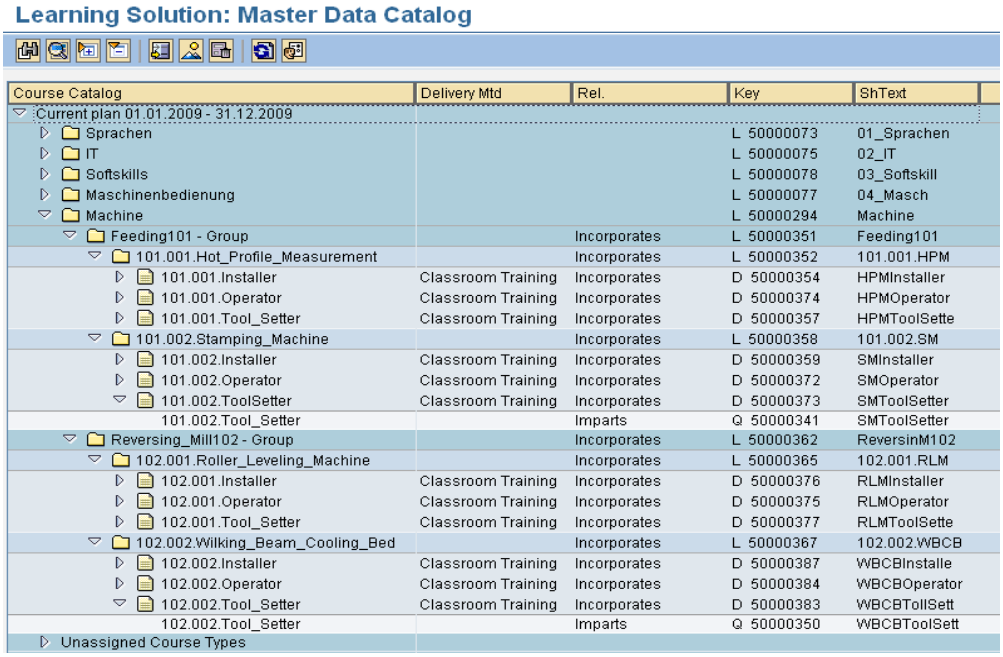


Fig. 4.9 Catalog de cursuri creat pentru mașinile de test

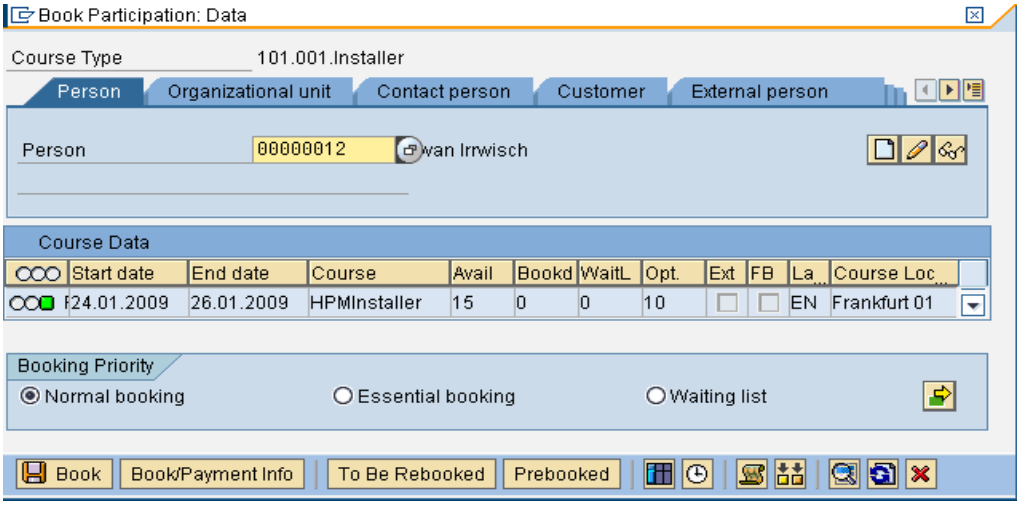


Fig. 4.10 Exemplu de rezervare a unui curs

Asignarea calificativelor angajaților se va face de către administrator după absolvirea unui curs, sau în momentul în care o persoană este nou angajată în cadrul firmei și dispune de anumite cunoștiințe. Astfel, administratorul poate asigna angajatului calificative pe baza diplomelor pe care acesta le are. În Fig. 4.11 se prezintă calificativele asignate subiectului cu ID-ul (Personal Number) 9.

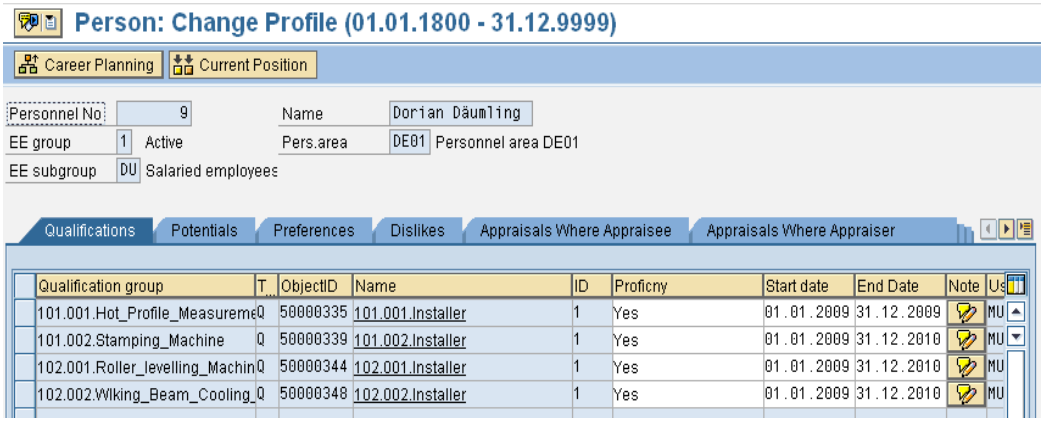


Fig. 4.11 Exemplu de calificative asignate subiectului cu ID 9

4.3 Structura bazei de date creată pentru integrarea obiectelor protejate

Pentru stocarea datelor necesare QBAC s-au folosit baze de date relaționale [71], iar pentru lucrul cu aceste date s-a folosit SQL. Deoarece declarațiile DDL de genul create, alter sau drop nu sunt suportate, s-au folosit uneltele puse la dispoziție de către ABAP Dictionary [72], [73] pentru a crea toate elementele de dezvoltare necesare realizării bazei de date corespunzătoare.

Pentru a crea obiectele de dezvoltare în ABAP Dictionary se pun la dispoziție o largă varietate de opțiuni, ca de exemplu tranzactia SE11 (Dictionary Maintenance), sau meniul contextual al pachetului în care dorim să creăm acele obiecte. Astfel, există posibilitatea de a crea de la tabele și elemente de date globale până la ajutor pentru căutare (search help) și views. Totodată, se dispune de o largă varietate de unelte pentru activarea și ajustarea tabelelor (ex. transacția SE14) și până la unelte pentru SQL- trace (ex. tranzacția ST05) în vederea optimizării declarațiilor SQL folosite la lucrul cu baza de date. De asemenea de un real folos poate fi și Data Modeler (ex. tranzacția SD11).

În cadrul ABAP Workbench s-a creat un nou pachet numit Y_NASAPCFRFID_DATABASE unde s-au realizat toate obiectele de dezvoltare necesare: tabele, views, tipuri de dată globale, domenii, tip de dată tabelă, ajutor pentru căutare de genul search help. În Fig. 4.12 se prezintă structura pachetului ce conține obiectele de dezvoltare.

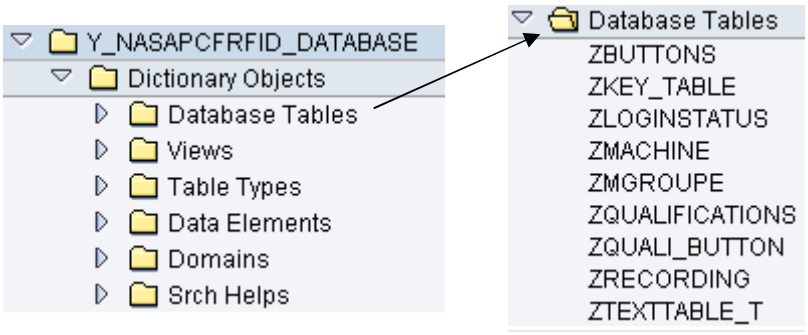


Fig. 4.12 Structura pachetului creat pentru dezvoltarea bazei de date

La rândul lui, fiecare folder conține obiectele ce sunt grupate în categoriile respective. De exemplu folderul:

- **Database Table** conține tabelele: ZLOGINSTATUS, ZGOUPE, ZMACHINE, ZBUTTONS, ZQUALIFICATIONS, ZQUALI-BUTTON, ZRECORDING, ZKEYTABLE și ZTEXTTABLE_T. Tabelele ZKEYTABLE și ZTEXTTABLE_T au fost create pentru a se putea stoca datele multilinguale în baza de date (pentru descriere în mai multe limbi). Folosind o extra tabelă așa numită tabelă de chei și o extra tabelă, așa numită tabelă de text se vor putea stoca în mai multe limbi descrierile text aferente butoanelor, grupelor și mașinilor.

- **Views** conține 12 views, necesare selectării datelor ce sunt împărțite în mai multe tabele. O însemnătate deosebită o are view-ul Z_HRP1001 prin intermediul căruia s-a realizat legătura cu datele din LSO, Personal Development. S-a realizat această legătură prin intermediul unui view, deoarece s-a dorit să se selecteze doar acele obiecte de tip Q care sunt importante scopului propus.

- **Domains** conține toate domeniile create pentru scopul QBAC. Un domeniu restricționează valorile care pot fi inserate într-un câmp al unei tabele, având în acest caz doar valori fixe prestabilite. Un domeniu poate să folosească ca și valori pentru restricționare: valori simple, intervale (ex. 40-100) sau o tabelă de valori în cazul în care se dispune de o largă gamă de valori.

- **Data elements** conține toate tipurile de dată globală create.

- **Srch Helps** conține serch help simple și complexe, create și atașate tabelelor corespunzătoare în așa fel încât administratorul nivelului celui mai de jos să poată să beneficieze de F4 help la căutarea anumitor valori în tabele. În acest sens am creat un număr de 18 search help.

Baza de date creată în ABAP Dictionary și conexiunile aferente sunt prezentate în Fig. 4.13.

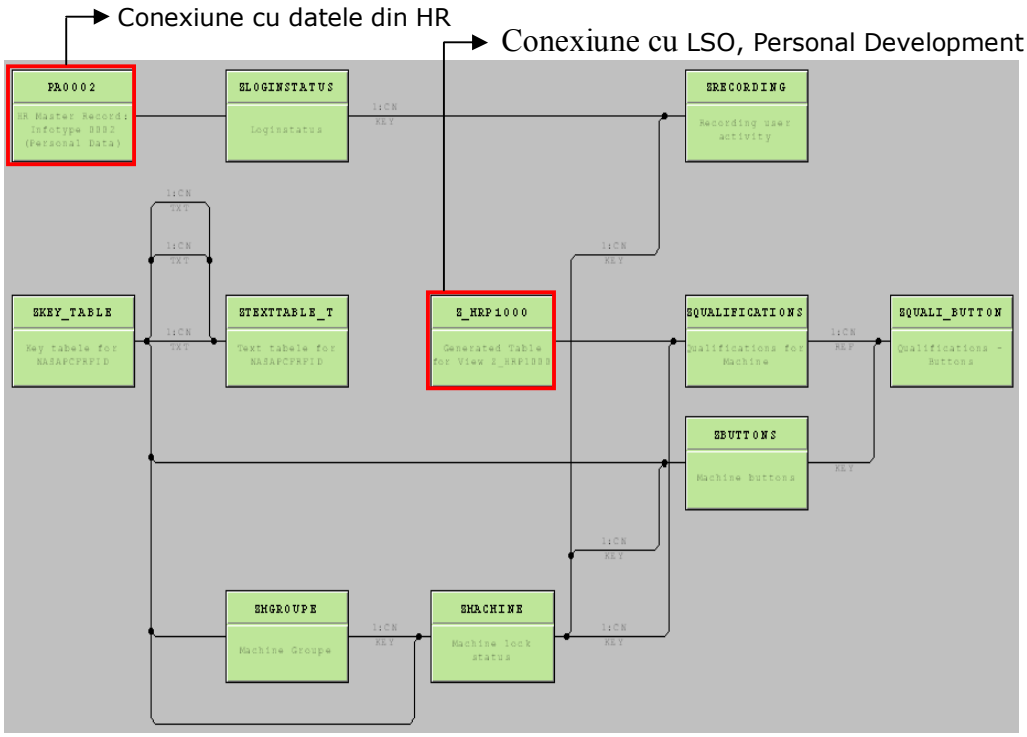


Fig. 4.13 Structura bazei de date create în ABAP Dictionary

Structura tabelelor cu care s-a interconectat baza de date creată, în vederea integrării obiectelor protejate și a datelor din HCM au fost prezentate în capitolul anterior. Schematic această conexiune este rerezentată în Fig. 4.14.

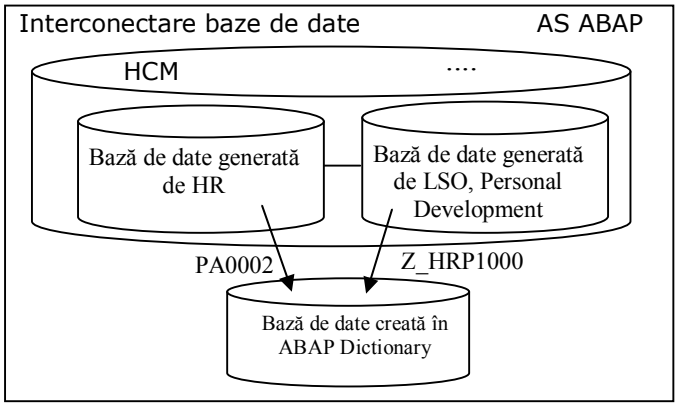


Fig. 4.14 Reprezentare schematică a conexiunilor cu datele HCM

Așadar, baza de date creată extinde vasta bază de date a HCM, atașând datelor angajaților și calificativelor noua funcționalitate, care se va materializa apoi în metoda de acces controlat QBAC.

Pentru a procesa datele din baza de date creată cu ajutorul ABAP Dictionary se va folosi SQL – DML de genul insert, update, delete, modify, select, iar consistența datelor vehiculate este oferită de conceptul SAP-LUW.

Conceptul ABAP LUW ne asigură că dacă, dintr-un program sau aplicație se modifică date într-o tabelă din ABAP Dictionary nici un alt program sau aplicație nu va putea modifica aceleași date până când nu s-a terminat prima sesiune de modificare, Fig. 4.15 (adaptată din [74]).

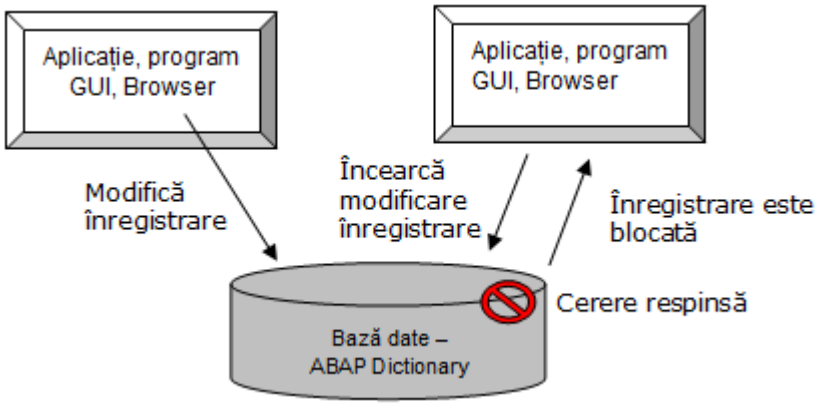


Fig. 4.15 Schematizare concept ABAP LUW pentru varianta de modificare simultană a unei înregistrări într-o tabelă ABAP Dictionary

4.4 Crearea logicii modelului de acces controlat

Motivele alegerii AS ABAP, s-au precizat la începutul acestui capitol. Pentru a crea toate obiectele de dezvoltare necesare s-a folosit ABAP ca și limbaj de programare împreună cu uneltele puse la dispoziție de către ABAP Workbench. În Fig. 4.16 se prezintă structura de pachete folosite pentru stocarea obiectelor respective.

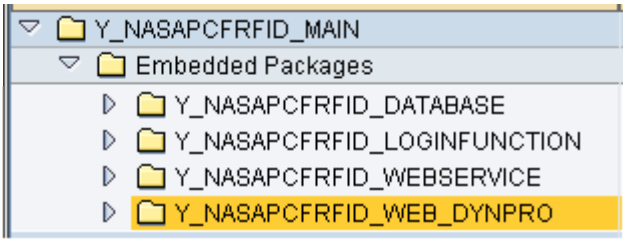


Fig. 4.16 Pachete folosite pentru obiectele de dezvoltare necesare

Pachetul ***Y_NASAPCFRFID_LOGINFUNCTION*** conține toate obiectele necesare realizării logicii modelului de acces controlat, pachetul ***Y_NASAPCFRFID_WEBSERVICE*** conține funcționalitatea oferită ca un web service, pachetul ***Y_NASAPCFRFID_WEB_DYNPRO*** conține toate elementele de dezvoltare create pentru realizarea interfeței de administrare (Web Dynpro ABAP) al nivelului de interfață cu obiectele protejate, iar conținutul pachetului ***Y_NASAPCFRFID_DATABASE*** a fost prezentat în subcapitolul anterior.

În continuare se va prezenta structura câtorva dintre obiectele create, împreună cu anumite elementele de limbaj ABAP utilizate.

4.4.1 Elementele de limbaj ABAP utilizate

ABAP [73], [74] [75], [76] este limbajul proprietar dezvoltat de către SAP pentru realizarea de aplicații comerciale, oferind o serie de ***avantaje*** ca de exemplu [43]:

- **suportă OOP** necesar pentru: folosirea noilor tehnologii (ex. Web Dynpro ABAP), pentru conceptul de moștenire simplă, interfețe, polimorfism, posibilitate de lucru cu evenimente, shared objects, etc;
- prezintă **funcționalități integrate** ca de exemplu: unelte pentru optimizare, SQL, SAP LUW, XML;
- suportă **programare procedurală** din motive de compatibilitate;
- suportă o **largă varietate de declarații** ce pot fi folosite (ex. field-symbols, call transaction, authority-check, raise event);
- oferă suport multilingual.

Pentru realizarea logicii necesare QBAC au fost necesare o serie de opearații de manipulare a datelor din baza de date. Cu ajutorul limbajului ABAP se pot accesa datele din baza da date ABAP Dictionary atât folosind declarații open SQL, cât și un acces obiect-orientat prin care Object Services [79] [80] [81] vor realiza aceste operații, Fig. 4.17.

În crearea logicii QBAC, s-a lucrat foarte mult cu tabele interne (internal table) și câmpuri symbol (field symbols) necesare pentru prelucrarea internă a datelor preluate din tabele create în ABAP Dictionary.

Pentru accesarea datelor din tabele interne și modificarea corespunzătoare a acestora s-a folosit de cele mai multe ori LOOP AT…. ENDLOOP, Fig. 4.18.

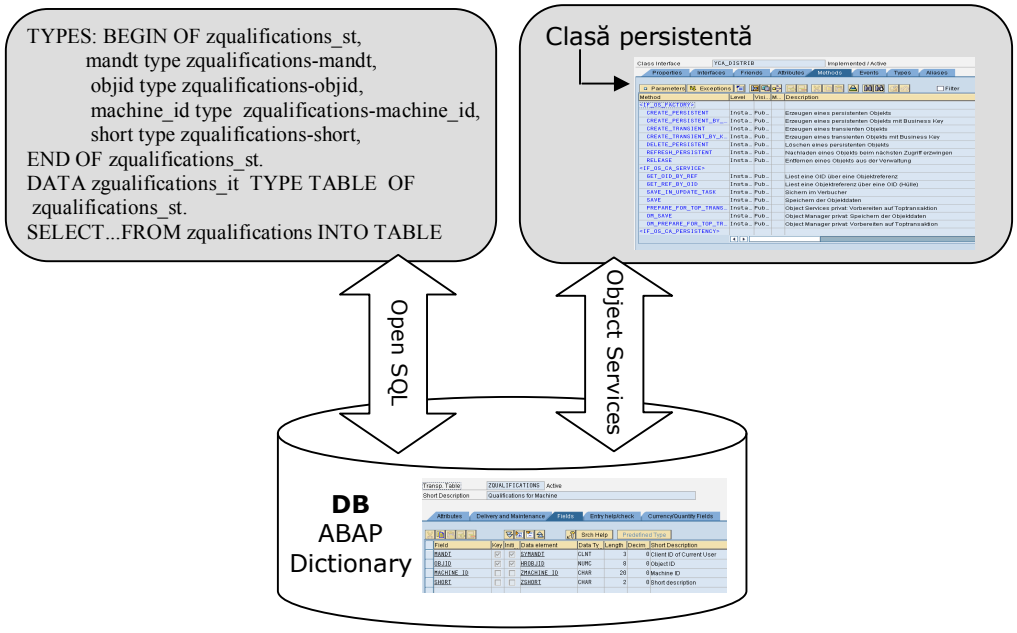


Fig. 4.17 Posibilități de accesare a datelor din ABAP Dictionary

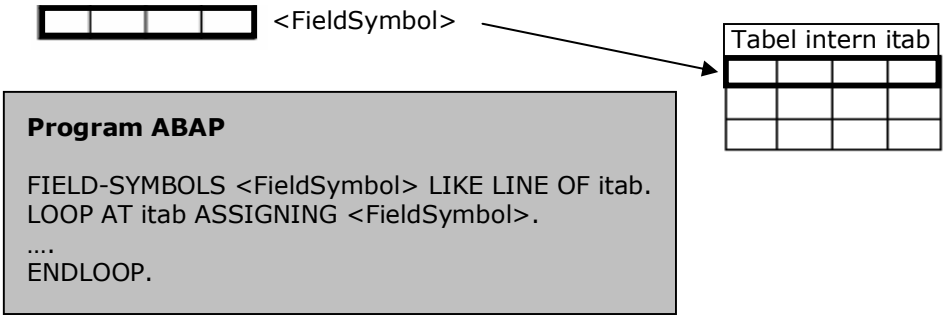


Fig. 4.18 LOOP AT… ENDLOOP și field symbols

În acest mod s-au citit secvențial liniile tabelului intern folosind LOOP AT și s-au realizat modificările necesare, sau s-a verificat dacă o anumită valoare îndeplinește o condiție specificată. Astfel, componentele unei linii au fost asignate la field symbol <Field_Symbol>.

Pentru monitorizarea programelor realizate, ABAP Workbench pune la dispoziție o largă gamă de unelte. În acest scop se va face o analiză folosind tranzactia SE30 (ABAP Runtyme Analysis). Astfel, s-au creat două versiuni a unui program, unul în care am folosit SQL şi work area și unul în care s-a folosit SQL şi field symbols, Fig. 4.19.

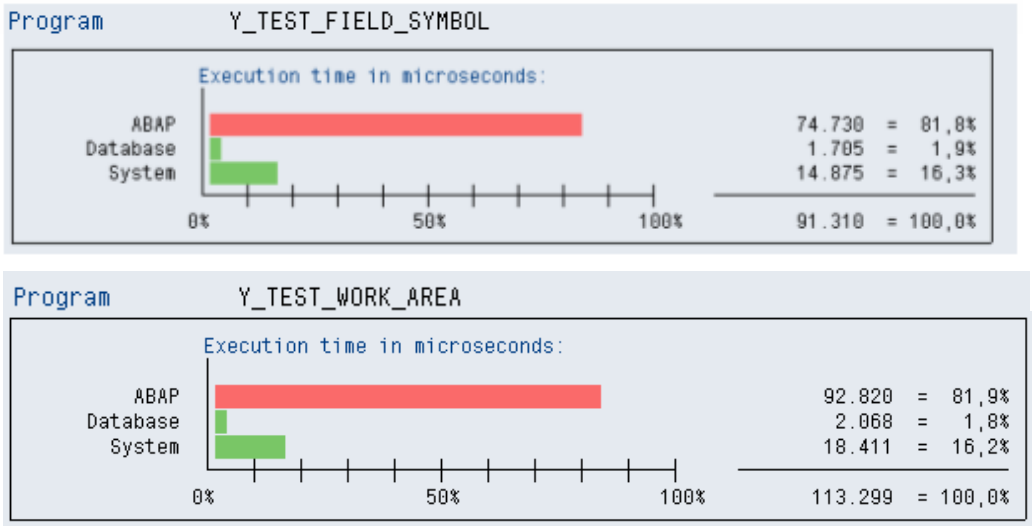


Fig. 4.19 Monitorizare performanțe pentru două programe de test

Pentru testarea și analizarea codului ABAP realizat pentru scopul proiectului de față, s-au folosit uneltele:

- **ABAP Debbuger** – care s-a folosit în cea mai mare parte pentru a executa programele realizate linie cu linie, sau secțiune cu secțiune. Astfel s-au putut detecta și repara erorile logice.
- **Syntax Check** – cu care s-a verificat dacă codul realizat este corect din punct de vedere al sintaxei. În cazul în care apar neconcludențe programul nu va fi executat, dar se va dispune de sugestiile și informațiile necesare astfel încât să se poată corecta erorile respective.
- **Code Inspector -** folosit pentru verificare sintaxă, securitate, performanță.
- **ABAP Runtime Analises** - care s-a pus în evidență prin studiul de performață prezentat anterior.

Pentru realizarea aplicațiilor sigure la nivelul server, s-au realizat următoarele ***operații de bază***:

- **filtrarea intrărilor** de la utilizator, chiar dacă acești utilizatori sunt proprii angajați, înlăturând SQL injection;
- **verificarea autorizării utilizatorilor**, inclusiv al celui cu care se face conexiunea prin intermediul Web Service, înlăturând eventualele backdors;

- prin **folosirea tehnologiei Web Dynpro ABAP** se reduce numărul eventualelor atacuri ce se pot realiza asupra unei tehnologii UI, conform [82].

4.4.2 Structura claselor și a metodelor create pentru login și logout

Pentru implementarea QBAC s-a folosit atât programarea orientată pe obiecte cât și programarea clasică, îmbinând cele două modele de programare din motive bine întemeiate. Pentru realizarea logicii QBAC s-au creat clase și metode folosind ABAP Objects, iar pentru oferirea sesiunilor de logare și delogare prin intermediul unui web service a fost necesară crearea unui Function Module (ca și end-point), folosind astfel programarea clasică. Această îmbinare a fost necesară datorită faptului că un Web Service (de tipul inside-out) poate fi creat folosind ABAP Workbench doar dintr-un Function Module, Function Group, BAPI și Interface message. Se putea folosi componenta SAP XI [83] și să nu se apeleze la programarea clasică, dar ținând cont de faptul că este nevoie de un singur Web Service folosirea unei noi componente SAP pentru o singură funcționalitate nu se justifică.

Folosind ABAP Objects s-au creat:

- clase globale vizibile pentru toate elementele de dezvoltare ale AS ABAP. Acestea au fost create cu ajutorul Class Builder;
- metode publice și private;
- parametrii de diferite tipuri, atribute;
- superclase, subclase.

Structura de clase realizate pentru funcționalitatea de login și logout pe partea de server este prezentată în Fig. 4.20. Această structura este identică cu cea prezentată în diagrama de clase a pattern-ului QBAC.

În **Anexa A2** se prezintă codarea unei metode, și anume a metodei **Qualifications** din clasa **YCL_LOGIN_SESSION**.

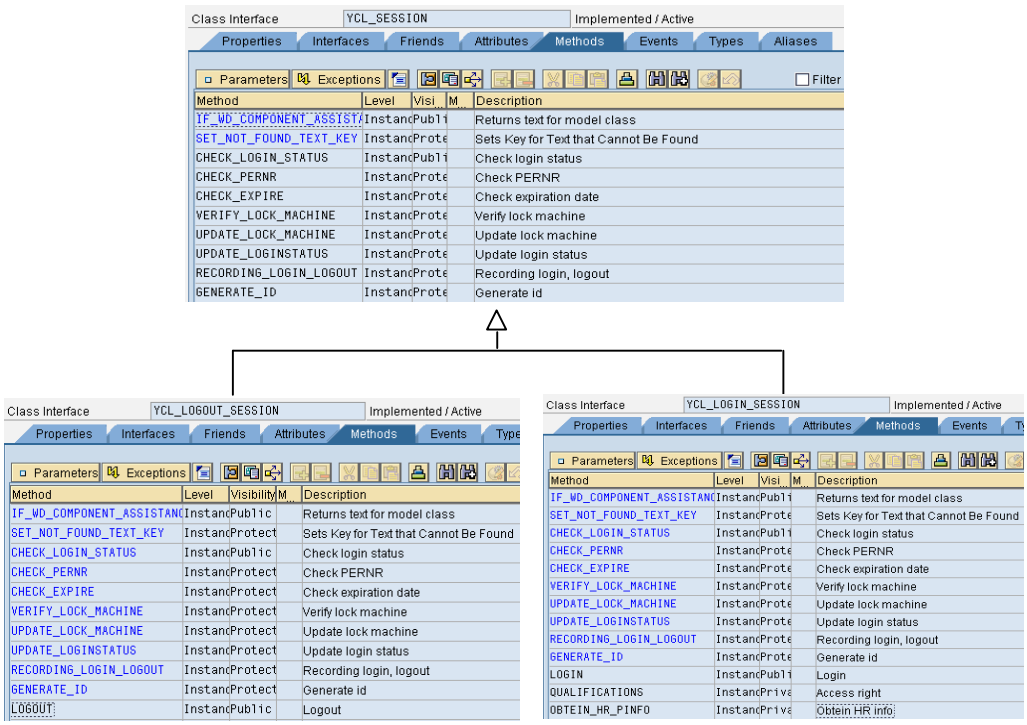


Fig. 4.20 Structura de clase pentru implementarea logicii de bază QBAC

4.4.3 Function Module creat

După cum s-a menționat, s-a creat un Function Module cu scopul de a-l transforma mai târziu într-un Web Service.

Un Function Module face parte din programarea clasică și nu poate fi creat de sine stătător decât într-un Function Group. Acestea reprezintă în esență programe ABAP de un anumit tip, care sunt foarte folosite pentru programarea SAP GUI. Pentru a crea un Function Module s-a folosit meniul contextual al pachetului Y_NASAPCFRFID_LOGINFUNCTION.

Cu ajutorul Function Module-ului creat se oferă posibilitatea de logare și delogare în/din sistem, apelând astfel metodele publice corespunzătoare claselor create, și care au fost prezentate anterior. Indiferent dacă logarea și delogarea se face prin intermediul unui Web Service, sau direct din ABAP (prin comunicare ActiveX cu PLC) Function Module va fi folosit ca și bază pentru logare și delogarea în, sau din sistem. În Fig. 4.21 se prezintă o parte din codarea Function Module-ului creat.

Acesta este un Function Module de tip Remote-Enabled Module, în consecință nu va suporta folosirea claselor de excepții (în tab-ul Exceptions) pentru tratarea erorilor, dar îndeplinește cerințele de folosire ca și punct final pentru un Web Service.

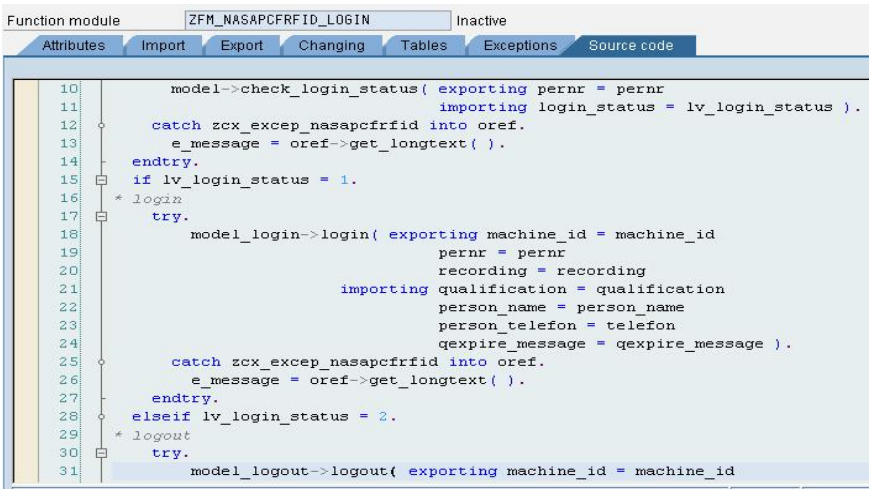


Fig. 4.21 Structură Function Module creat

4.4.4 Clase de excepții și mesaje create

Ca și alte limbaje de programare, limbajul ABAP oferă atât suport pentru tratarea excepțiilor din clase de excepții [84], cât și de tratare a excepțiilor care nu sunt bzate pe principiul claselor. De asemenea oferă și posibilitatea de a folosi diferite tipuri de mesaje. În acest mod se pot crea aplicații ce tratează excepțiile, informează utilizatorul prin intermediul mesajelor neblcându-se la apariția diverselor tipuri de excepții.

Și în cazul acestei aplicații am folosit mesaje și excepții ce sunt create în clase corespunzătoare, sau stocate ca și texte în clase de asistență. Atunci când se folosește principiul OOP și în cazul excepțiilor, beneficiem de o serie de **avantaje**, ca de exemplu:

- suport pentru crearea aplicațiilor multilinguale. Astfel, stringurile conținute trebuie doar traduse, fără a necesita modificare în cod;
- centralizarea tuturor stringurilor folosite;
- folosirea principiului moștenirii.

Toate clasele de excepții din ABAP au ca și superclasă cx_root, dar o clasă de excepție creată poate moșteni din clasele cx_static_check, cx_dynamic_check, sau cx_no_check și nu direct din clasa cx_root, Fig. 4.22.

Pentru tratarea excepțiilor au fost create două clase de excepții globale, una pentru logica QBAC - ZCX_EXCEP_NASAPCFRFID și una pentru aplicația de administrare a nivelului celui mai de jos - YCX_EXCEPTION_ADMIN. Pentru crearea acestora am folosit tranzacția SE24 (Class Builder). O clasă de excepții globală se crează în ABAP la fel ca și o clasă normală, optându-se apoi pentru varianta clasă de excepții și precizându-i super clasa.

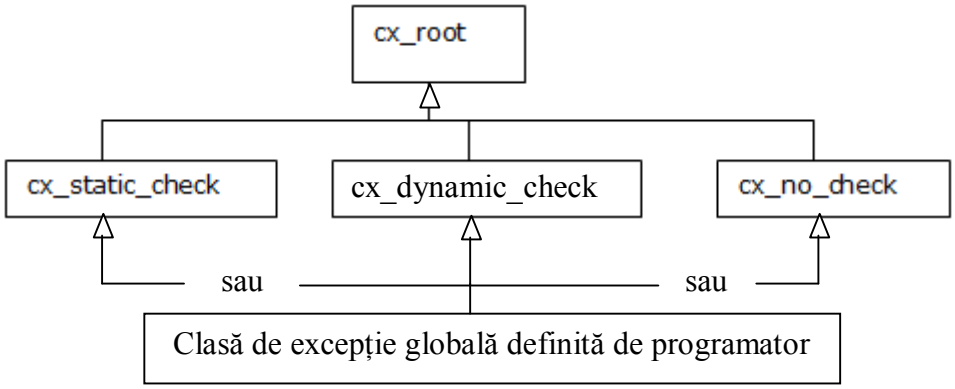


Fig. 4. 22 Superclase pentru clase de excepții ABAP

În Fig. 4.23 se prezintă câteva dintre excepțiile create în clasa de excepții globală ZCX_EXCEP_NASAPCFRFID.

Class Interface ZCX_EXCEP_NASAPCFRFID Implemented / Active

Properties | Interfaces | Friends | Attributes | Texts | Methods | Events

Long Text

Exception ID	Text
CX_ROOT	An exception occurred
ZCX_EXCEP_NASAPCFRFID	
ZCX_ID_DONT_EXIST	The personal number &ID& doesnt have the authorization!
ZCX_SELECT_LOCK_MACHINE	The lock machine selection cannot be done!
ZCX_SELECT_HR_DATA	Select name for pernr &ID& cannot be done!
ZCX_TELEFON_HR_DATA	Select phone number for pernr &ID& cannot be done!
ZCX_ID_HR_EXIST	The personal number &ID& is not found in HR!
ZCX_UPDATE_LOGIN_STATUS	Update login status cannot be done!
ZCX_UPDATE_LOCK_MACHINE	Update lock machine cannot be done!
ZCX_ERROR_ID	ID for recording data cannot be done!
ZCX_INSERT_LOGIN_DATA	Login or logout data cannot be done!
ZCX_MACHINE_IST_LOCK	The machine is locked!
ZCX_MACHINE_OR_USER	Cannot perform logout! The machine or the user is unlocked!
ZCX_SELECT_NO_POSSIBLE	Selecting the qualification - machine is not possible!

Fig. 4.23 Prezentarea câtorva dintre mesajele de excepții create

Pentru stocarea mesajelor necesare au fost folosite texte create în clase de asistență și în clase de mesaje, atât pentru partea de administrare, cât și pentru partea de logare și delogare. O clasă de asistență este practic o clasă normală ABAP, care are ca și superclasă clasa CL_WD_COMPONENT_ASSISTANCE. Avantajele folosirii clasei de asistență vor fi prezentate în paragraful următor.

Pentru a crea o clasă de mesaje se poate folosi tranzacția SE91. Mesajele stocate într-o clasă de mesaje dispun de un număr format din trei cifre și de o descriere aferentă. Aceste mesaje sunt stocate apoi în tabela T100, ajutând în comunicarea cu utilizatorul. Mesajele create în clasele de mesaje pot fi folosite direct în ABAP, sau pot să fie folosite pentru crearea textelor exceptiilor.

În **Anexa A3** se prezintă un exemplu de clasă de mesaje creată pentru aplicația de administrare, mesaje care vor fi folosite pentru formarea textelor din clasa de excepții YCX_EXCEPTION_ADMIN.

4.5 Aplicație de administrare Web Dynpro ABAP

Pentru administrarea nivelului celui mai de jos al QBAC, a fost nevoie să se creeze o aplicație de administrare în Web Dynpro ABAP. Prin intermediul acesteia, se oferă administratorului acestui nivel posibilitatea de a dispune de anumite funcționalități, ca de exemplu: posibilitatea de creare al diverselor rapoarte, de introducere al noilor grupe de mașini, de noi mașini împreună cu intrările acestora, posibilitatea de a introduce în logica QBAC a noi angajați, de a înlătura angajați din logica QBAC sau posibilități de căutare avansată.

În continuare se va prezenta structura acestei aplicații, precum și motivele alegerii Web Dynpro ABAP în defavoarea altor tehnologii.

4.5.1 Web Dynpro ABAP, funcționalități folosite, avantaje

După cum s-a putut observa și în Fig. 4.2 Web Dynpro ABAP [83], [84], [85], [86] face parte din nivelul de prezentare al AS ABAP. Pe lângă Web Dynpro ABAP, în zona de prezentare pentru AS ABAP mai dispunem de BSP și de SAP GUI. Cu ajutorul SAP GUI nu pot fi create aplicații web, aceasta fiind tehnologia SAP, ce a fost dezvoltată pentru a oferi interfețe grafice clasice numite și Dynpros. Cu ajutorul BSP se pot crea aplicații web, aceasta fiind tehnologia dinaintea Web Dynpro ABAP, ce combină ABAP și HTML, dispunând însă de multe dezavantaje și o structură complexă.

De aceea s-a optat pentru folosirea Web Dynpro ABAP, care este noua tehnologie SAP, cu ajutorul căreia se pot crea aplicații web „state of the art" bazate pe principiul MVC. Folosind această tehnologie dispunem de o serie de **avantaje** ca, de exemplu:

- **WYSIWYG** wie editor;
- **componentizarea** aplicațiilor realizate [89];
- **reutilizarea componentelor** realizate;
- **componente standard** (ex. SO - WDR_SELECT_OPTIONS, ALV - SALV_WD_TABLE) ce pot fi folosie pentru ușurarea muncii de programare;
- diferite metode standard, așa numitele **metode Hook**, cu ajutorul cărora se poate interveni la anumite momente în execuția programului [90];
- **programare statică și dinamică**. Fiecare element UI Web Dynpro dispune de o clasă care oferă posibilitatea realizării acestuia în mod dinamic [91];

- **nu necesită** cunoștințe de **HTML** sau **JavaScript**, este suficient cunoașterea limbajului ABAP și a modului de lucru cu Framework-ul Web Dynpro. Algoritmi complecși vor transforma apoi aplicația la momentul rulării în cod HTML, JavaScript sau XML;
- largă varietate de **elemente UI** care se pun la dispoziție [92];
- posibilitatea de **personalizare** a aplicațiilor realizate;
- folosirea **noilor tehnologii** ca de exemplu formularele Adobe [93];
- **numărul de atacuri** ce pot fi realizate asupra unei tehnologii UI sunt **reduse** în cazul folosirii Web Dynpro, conform [40].

În cadrul aplicației create a fost nevoie de realizarea diferitelor rapoarte pentru obținerea informațiilor legate de mașini, de activitatea angajaților, etc. Pentru realizarea acestor rapoarte nu a fost suficientă afișarea datelor în tabele fiind necesară afișarea datelor în format PDF, alături de posibilitatea de salvare a acestor rapoarte, cu data și ora corespunzătoare, astfel încât să poată fi arhivate.

Pentru a avea posibilitatea de a realiza rapoartele în format PDF s-a folosit SAP Interactive Forms by Abobe, care oferă posibilitatea de a crea atât formulare statice cât și formulare interactive. Pentru scopul implementarea QBAC a fost nevoie de varianta rapoartelor non-interactive.

Pentru design-ul formularelor Adobe s-a folosit Adobe LiveCycle Designer care oferă funcționalități complexe, având totodată posibilitatea de a mapa nodurile și atributele din Web Dynpro ABAP la elementele din formular. Elementul UI din Web Dynpro ABAP folosit pentru a lucra cu formulare Adobe este InteractiveForm. În Fig. 4.24 se prezintă un astfel de raport creat de către administrator.

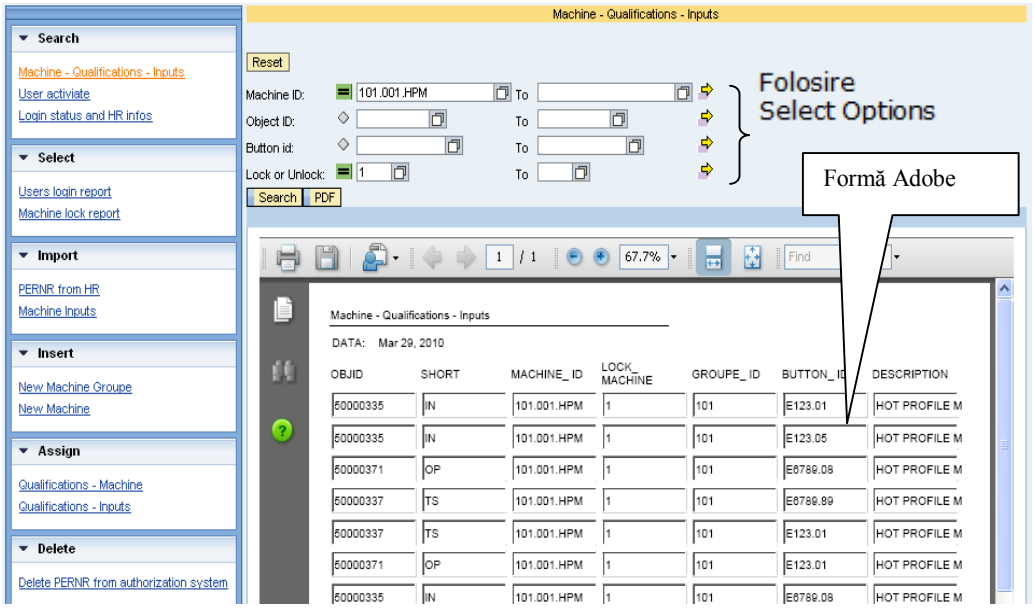


Fig. 4.24 Exemplu de raport creat

Astfel, se va putea genera un raport pe baza condițiilor de căutare, care va fi afișat în tabele folosid componenta ALV, sau se va putea afișa în format PDF ușor de savat și tipărit. Pentru căutare avansată s-a folosit componenta SO, cu ajutorul căreia se poate realiza dinamic ecranul de selectare și se beneficiază de diverse opțiuni de căutare (ex. valori în afara unui interval, valori mai mici decât o valoare dată).

Pentru realizarea diferitelor grafice s-a folosit elementul UI BusinessGraphics, pentru realizarea meniului s-a folosit elementul UI ContextualPanel, pentru realizarea pașilor wizard-urilor (necesari pentru importarea anumitor date și ștergerea anumitor date) s-a folosit elementul UI RoadMap, alături de alte elemente UI ca: Table, InpuFields, TextViews, și altele.

Importarea datelor în logica QBAC este folosită pentru:

- importarea în conceptul de autorizare QBAC a angajaților doriți, putând astfel realiza o importare în masă a angajaților, sau importarea doar a unui anumit angajat, sau doar a acelora ce îndeplinesc anumite condiții;
- importarea dintr-un fișier text a denumirii mașinilor și a intrărilor acestora.

Informații detaliate legate de Web Dynpro ABAP, elemente UI și exemple ale folosirii acestora se pot găsi la referința [92].

4.5.2 Structura aplicației de administrare realizată

După cum s-a precizat, toate elementele de dezvoltare necesare aplicației de administrare Web Dynpro ABAP au fost create în pachetul Y_NASAPCFRFID_WEB_DYNPRO. Aplicația este creată din 12 componente secundare ce au fost apoi introduse în componenta principală, Fig. 4.25.

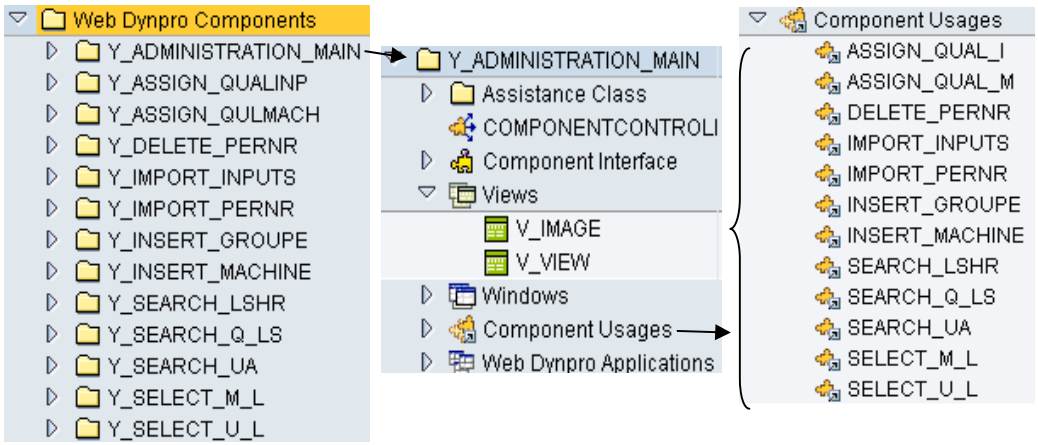


Fig. 4.25 Structura componentizată a aplicației create

Ca și model pentru aplicația creată s-a folosit o clasă de asistență (Anexa A4) ce conține un număr de 26 de metode. Dintre **avantajele** folosirii unei **clase de asistență** ca și model în Web Dynpro ABAP se pot menționa [94]:

- folosirea metodelor claselor de asistență sunt mai eficiente decât folosirea metodelor din cadrul componentelor;
- oferă suport pentru crearea așa numitelor text symbols ce au fost folosite pentru crearea mesajelor multilinguale;
- clasa de asistență este automat instanțiată, accesarea metodelor sale făcânduse cu ajutorul atributului wd_assist (referință la instanța clasei de asistență).

O componentă Web Dynpro este formată din mai multe părți (ex. view, window), fiecare dintre acestea având anumite metode Hook prestabilite, în care putem introduce cod ABAP, pentru a interveni în anumite momente în execuția aplicației. Dintre acestea, cele mai folosite metode Hook, în cazul aplicației realizate sunt:

- **wdDoInit()** care este considerată metodă constructor, și care a fost folosită pentru operații de inițializare.
- **wdDoOnContextMenu()** folosită pentru a crea meniul contextual (clic dreapta) pentru aplicație.
- **wdDoBeforeAction()** folosită pentru realizarea propriilor verificări înainte ca o acțiune să fie realizată.

Pe lângă acestea se dispune și de alte tipuri de metode, ca de exemplu metode supply, metode ce tratează un eveniment (event handler), sau metode definite de programator. Întreaga aplicație creată dispune de un număr de 93 metode definite de programator. În **Anexa A5** se prezintă codarea realizată într-o metodă definită de programator și event handler.

Așadar, structura three – tier rezultată pentru aplicația creată, conform MVC este formată din view, ce reprezintă nivelul de interfață, model responsabil cu procesarea datelor și controller responsabil cu evaluarea cererilor, trimiterea datelor și instrucțiunilor la model, comunicarea cu componentele view, Fig. 4.26.

Ca și model pentru o aplicație Web Dynpro (pe lângă clasa de asistență folosită), se pot folosi o serie de componente, ca de exemplu:

- clasă ABAP;
- un web service;
- un BAPI;
- o componentă Web Dynpro faceless (nu dispune de părți vizuale).

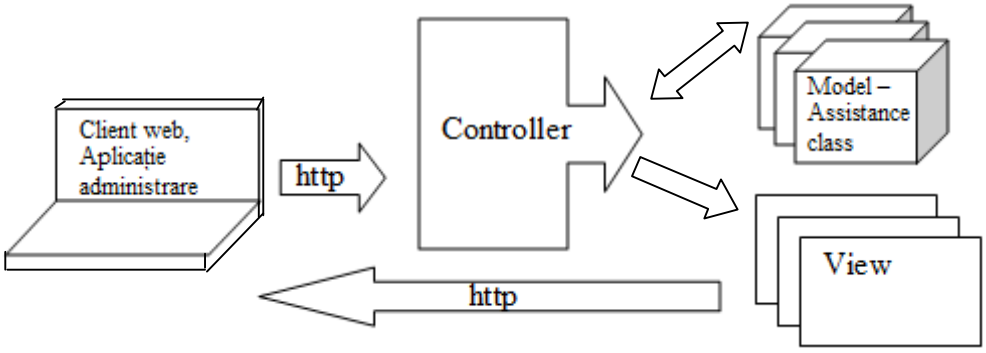


Fig. 4.26 Structura tree – tier rezultată

4.6 Web Service-ul de tip inside-out folosit

Cu ajutorul ABAP Workbench se poate crea și folosi un Web Service, ABAP Workbench putând fi astfel atât un Web Service provider cât și un Web Service consumer. Folosirea de Web Services [95], [96] este din ce în ce mai frecventă, începând de la funcționalități de genul E-comerce și ajungând până la folosirea acestora în automatică [50].

În Fig. 4.27 se prezintă schematic structura Framework-ului Web Service cu AS ABAP (adaptat din [94]).

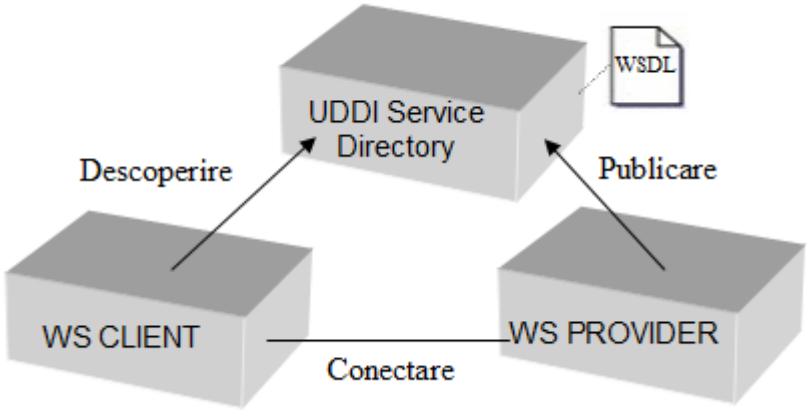


Fig. 4.27 Framework-ul Web Service cu AS ABAP

Un Web Service creat cu AS ABAP poate fi astfel publicat într-un UDDI, de unde va putea fi găsit (descoperit) de către clienții interesați. Publicarea Web Service-ului într-un UDDI nu este obligatorie, cel care oferă Web Service-ul poate informa direct consumatorul de Web Service, oferindu-i fișierul WSDL.

Dintre ***avantajele*** folosirii unui Web Service în cazul aplicației realizate menționăm:

- este disponibil cu ajutorul internetului sau intranetului (ethernet), putând astfel să fie foarte ușor consumat în aplicația Visual Basic (sau alt tip de aplicații) de la nivelul de control;

- independent de orice limbaj de programare folosind gramatica XML [96], [97], [98], [99] putând astfel fi folosit chiar și la nivelul PLC.

După cum s-a menționat, pentru implementarea QBAC s-a creat un ***Web Service*** de tipul inside-out numit ZNASAPCFRFID, care are:

- **intrări**: machineID (id-ul mașinii unde se face logarea sau delogarea), PERNR (id-ul subiectului care dorește logare sau delogare) și recording (eventuala activitatea a unui subiect la una dintre mașini – pentru cazul de delogare);

- **ieșiri**: PersonName (numele subiectului care se loghează), dreptAces (dreptul de acces codat într-un întreg), telefon (numărul de telefon al celui care se loghează), Smessage (Mesaj de succes în caz de delogare cu suces), Emessage (Mesaje de eroare) QExpireMessage (Mesaje în cazul în care unul sau mai multe calificative sunt pe cale să expire) și ExpireDate (dată de expirare pentru unul sau mai multe calificative).

Pentru securizarea Web Service-ului creat s-a folosit un user și o parolă urmând ca pentru viitoarele dezvoltări să se adauge semnături digitale. Mai multe detalii referitoare la securitatea Web Service-lor cu AS ABAP se pot găsi [101].

4.7 SAP NetWeaver Portal

SAP NetWeaver Portal [70], [102], [103] este componenta SAP cu ajutorul căreia se poate extinde QBAC cu RBAC astfel încât angajații să poată participa la E-learning, să poată avea acces la aplicații de genul ESS, oferind totodată administratorului nivelului celui mai de jos posibilitatea de a accesa aplicația Web Dynpro creată. Tot cu ajutorul SAP NetWeaver portal se poate crea portalul firmei.

Accesul in portal se face pe bază de roluri, iar ceea ce un user are posibilitatea de a vedea după logarea în portal depinde de rolurile care i-au fost asignate de către administrator. Accesarea componentelor ce nu fac parte din sistem se face cu ajutorul mecanismului SSO, conform căruia utilizatorul trebuie să se logheze o singură dată, iar după logare acesta primește un așa numit log-on ticket cu ajutorul căruia poate accesa celelalte componente, fără a fi nevoit să se logheze pentru fiecare aplicație separat. Detalii referitoare la conceptele de autorizare și autentificare SAP se pot găsi la [104].

Liderii pieței pentru 2008 în zona produselor portal sunt prezentați în Fig. 4.28. Mai multe detalii legate de acest grafic și motivele împărțirii în cele patru zone pot fi găsite la [63].

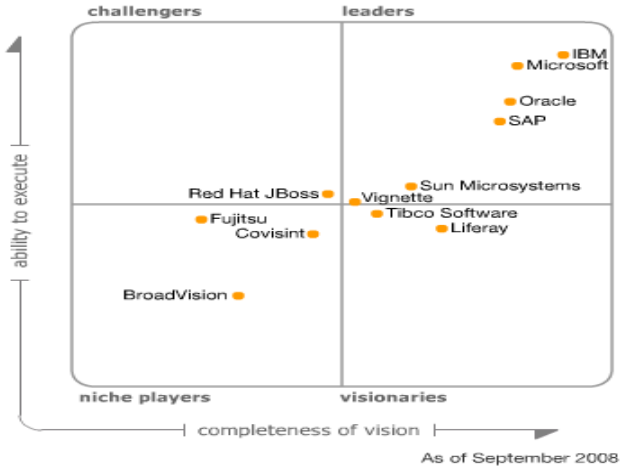


Fig. 4.28 Liderii pieței - Gartner Group 2008 [63]

După cum se poate observa, SAP face parte dintre liderii celor care oferă astfel de aplicații soft, fiind plasat în anul 2008 pe locul patru după IBM, Microsoft și Oracle. Din păcate din 2006 până în 2008 SAP a pierdut un loc în poziția celor mai importanți provideri de astfel de aplicații (conform aceluiași Gartner Group pentru 2006).

Cea mai mică unitate ce se poate crea în SAP NetWeaver Portal este un așa numit iView care s-a folosit pentru introducerea aplicației de administrare Web Dynpro creată. Într-un iView pot fi integrate o largă serie de obiecte, de la tranzacții și web services până la websites.

Avantajele folosirii portalului în cazul QBAC sunt:

- oferă posibilitatea combinării QBAC cu RBAC;
- pune la dispoziție diverse servicii (ex. Căutare) și aplicații ce pot fi utilizate;
- datorită SSO utilizatorul nu trebuie să memoreze o multitudine de parole;
- conținutul portalului poate fi creat cu diverse unelte, de la wizards până la unelte ce nu necesită cunoștințe de programare (ex. Visual Composer) și unelte de programare complexe (ex. Web Dynpro ABAP);
- diversele aplicații și obiecte create cu ajutorul diferitelor tehnologii (SAP sau non SAP) pot fi combinate, dând impresia unui tot unitar;
- posibilitatea angajaților de a-și manageria propriile date personale prin intermediul aplicațiilor de genul ESS (**Anexa A6**). Astfel datele din HR vor fi mereu actualizate fără a necesita eforturi adiacente din partea administratorului acestui nivel;
- comunicarea cu stack-ul ABAP pentru obținerea datelor necesarare este ușor de realizat, necesitând doar anumite setări.

Structura portalului cu rularea aplicației de administrare este prezentată în Fig. 4.29.

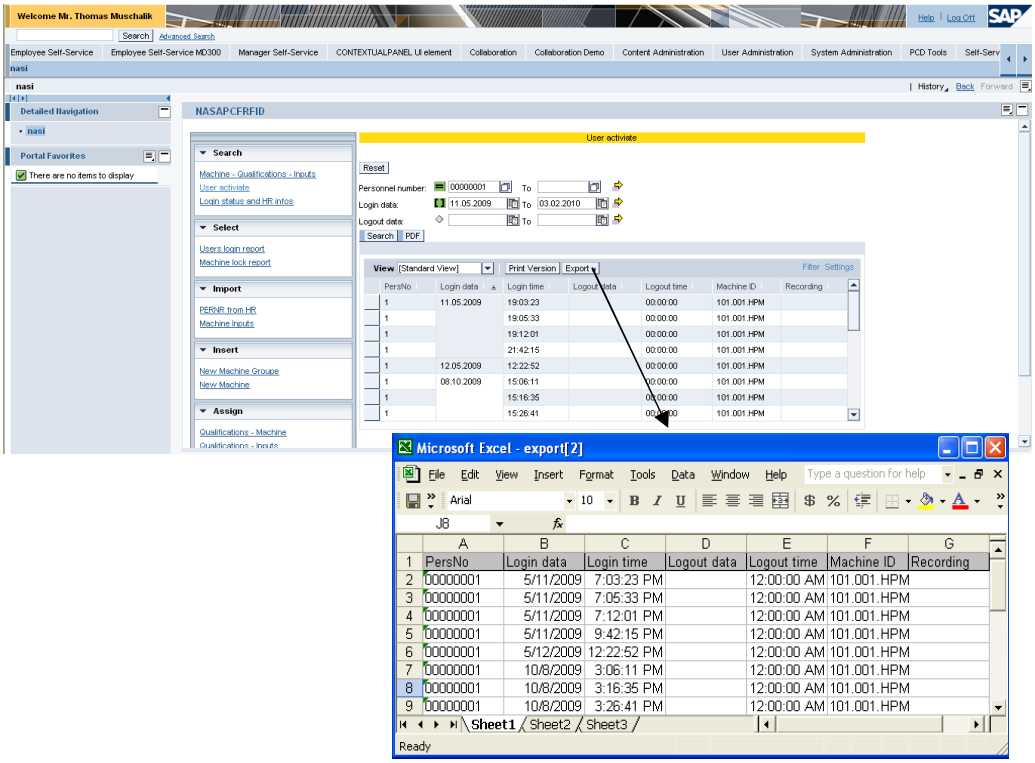


Fig. 4.29 Structura aplicației de administrare a interfeței cu mașinile rulând în portal

4.8 Internaționalizarea aplicațiilor

Platforma SAP NetWeaver oferă posibilitatea de creare al aplicațiilor multilinguale, care să ofere suport pentru limbile dorite fără a fi nevoie de recodarea acestora. În acest sens SAP pune la dispoziție o serie de unelte alături de care trebuie să îndeplinim anumite cerințe, ca de exemplu neintroducerea în cod a stringurilor specifice textelor ce se doresc a fi făcute multilinguale, deci traduse în mai multe limbi. De asemenea, pentru AS ABAP limbile pentru care se dorește a se oferi suport trebuie să fie instalate în sistem (tranzacția SMLT oferă informații în acest sens).

În cazul QBAC un rol deosebit au avut realizarea multilinguală cu suport pentru engleză și germană a următoarelor componente:

- datele înregistrate în baza de date;
- aplicația Web Dynpro;

- mesajele și excepțiile;
- catalogul de calificative și de cursuri;
- datele oferite către PLC.

Deoarece este nevoie de înregistrarea în baza de date a descrierilor mașinilor, a grupelor de mașini și a butoanelor fiecărei mașini, astfel încât acestea să fie oferite apoi în limba dorită, a fost necesar crearea a două extra tabele. După cum s-a putut observa în Fig. 4.13 s-au creat două extra tabele **ZKEYTABLE** și **ZTEXTTABLE_T** cu ajutorul cărora s-au putut apoi stoca stringurile respective în funcție de o anumită limbă. Stringurile introduse într-o limbă inițială (engleza în cazul de față), pot fi apoi traduse și în alte limbi (germana pentru cazul QBAC). Selectarea datelor se va face apoi în funcție de limba de logare a celui care dorește să vizualizeze acele date, realizând o selecție în funcție de câmpul de limbă SPRAS.

Așa cum se recomandă [105], în cazul proiectelor SAP cu suport pentru mai multe limbi trebuie să se stabilească inițial o anumită limbă și să se creeze toate elementele de dezvoltare în respectiva limbă (engleza cazul nostru). Apoi, va urma procesul de traducere a stringurilor respective în limbile pentru care se oferă suport. Un motiv al alegerii englezei ca și limbă inițială este faptul că dezvoltarea acestui proiect s-a realizat in Germania, iar limba germană nu este limba maternă a dezvoltatorilor.

Pentru realizarea mesajelor și excepțiilor multilinguale [106] s-au folosit clase de mesaje și excepții precum și stringuri introduse în clase de asistență. Cu ajutorul uneltelor puse la dispoziție de către ABAP Workbench s-au tradus stringurile respective urmând ca mesajele să fie apoi afișate în funcție de limba de logare.

Pentru a realiza multilingual aplicația Web Dynpro:

- s-a folosit OTR;
- s-au tradus textele de descriere al obiectelor de dată globale, domenii și alte obiecte de dezvoltare realizate în ABAP Dictionary;
- s-au folosit texte din clasa de asistență folosită ca și model;
- s-au folosit clasele de excepții și mesaje create;
- s-au selectat datele din baza de date în funcție de limba de logare;
- nu s-au folosit stringuri statice pentru definirea proprietăților elementelor UI folosite;
- s-au tradus textele din formele Adobe.

În ***Anexa A7*** se prezintă o captură din aplicația de administrare în care limba de logare este germana.

Catalogul de calificaive [107] se poate și el traduce, astfel încât să poată fi încadrat în procesul multilingual. Cu ajutorul uneltelor standard puse la dispoziție se oferă posibilitatea de a traduce stringurile corespunzătoare, afișarea acestora într-o limbă sau alta facându-se tot în funcție de limba de logare. În ***Anexa A8*** se prezintă catalogul de calificative pentru limba germană.

Așadar, urmând specificațiile SAP în domeniu s-au creat toate elementele de dezvoltare multilinguale. Oferirea de suport pentru alte limbi în afară de engleză și germană este ușor de realizat, fără a necesita recodare, ci doar o simplă traducere a unor stringuri centralizate.

Deciderea faptului dacă un proiect trebuie să acorde sau nu suport multilingual este una dintre întrebările la care trebuie răspuns încă din prima fază de dezvoltare a unui proiect. Dacă se ține cont de acest lucru întregul proces va decurge fără necesitatea de modificare a aplicațiilor realizate în cazul oferirii de suport și pentru alte limbi.

Concluzii:

În cadrul acestui capitol:

- s-au analizat avantajele prin care platforma de integrare și aplicații SAP NetWeaver ajută la implementarea pattern-ului QBAC. Astfel prin folosirea unei platforme de integrare se accelerează procesul de implementare, folosind acolo unde este posibil uneltele și modulele puse la dispoziție de către platorma respectivă, dispunând totodată de un sistem complex;

- s-au ales modulele și uneltele SAP pentru implementarea QBAC. Astfel, s-au ales modulele SAP NetWeaver Portal, SAP ERP HCM și s-a ales folosirea Application Server ABAP. Alegerea Application Server ABAP (ABAP ca și limbaj de programare) a fost realizată deoarece modulul SAP ERP HCM stochează datele și aplicațiile în acest server de aplicații. Folosirea Java ca și limbaj de programare ar fi necesitat o comunicare între Application Server ABAP și Application Server ABAP pentru obținerea datelor din HCM ceea ce ar fi dus la afectarea negativă a vitezei de execuție;

- s-au creat toate elementele de dezvoltare necesare implementării QBAC la nivel de server (platforma SAP NetWeaver): clase, Function Module, Web Service, aplicație de administrare, baze de date, etc.);

- s-au folosit uneltele standard SAP pentru internaționalizarea întregului proiect realizat, oferind suport pentru germană și engleză. Extinderi viitoare nu necesită recodare ci doar traducerea unor stringuri centralizate;

- s-a folosit programarea ABAP pentru implementarea algoritmilor necesari implementării logicii QBAC, dispunând astfel de o largă gamă de funcționalități ce pot fi folosite pentru ușurarea muncii de programare;

- s-a folosit tehnologia Web Dynpro pentru crearea aplicației de administrare al nivelului de interfață cu obiectele protejate. Web Dynpro este tehnologia SAP cu ajutorul căreia se pot crea aplicații web „state of the art" folosind principiul MVC. Astfel, s-a oferit administratorului acestui nivel posibilități avansate de căutare, posibilitate de creare al rapoartelor atât în format PDF ușor de salvat și arhivat cât și în varianta de exportare al datelor în Excel.

Concluzionând, se poate spune că, platforma de integrare și aplicații SAP NetWeaver a oferit uneltele necesare implementării QBAC, accelerând procesul de implementare și oferind totodată soluții complexe.

5. METODA DE AUTENTIFICARE UTILIZATĂ PENTRU IMPLEMENTAREA QBAC

În cadrul acestui capitol se face o scurtă trecere în revistă a diferitelor metode de autentificare existente, scoțând în evidență motivele alegerii cardurilor RFID pentru identificarea subiecților proiectului nostru. Metoda de autentificare folosită pentru implementarea QBAC nu face parte din calculul dreptului de acces. Așadar, alegerea unei alte metode de autentificare este posibilă fără a avea ca rezultat modificarea logicii QBAC (autentificarea și autorizarea fiind două problematici diferite, dar care fac parte din accesul controlat la resurse). În cazul în care se dorește schimbarea metodei de autentificare de la RFID la o altă metodă, schimbările ce trebuiesc realizate necesită doar introducerea noii metode de autentificare (ex. baze de date pentru stocarea datelor biometrice [108], hardware suplimentar). Calculul dreptului de acces (autorizării) va rămâne nemodificat. Pentru extinderi viitoare în vederea implementării productive, se iau în vedere colaborări viitoare cu firme (ex. PCS) ce oferă module specializate de autentificare.

5.1 Metode de autentificare

Prin intermediul procesului de autentificare se determină dacă o entitate este cea care pretinde a fi. Acesta este procesul desfășurat înaintea determinării nivelului de autorizare pe care respectiva entitate o are. Procesul de autentificare răspunde la întrebarea: *„Cine este entitatea ce dorește să acceseze resurse?”,* urmat de procesul de autorizare ce dă răspuns întrebării: *„Ce resurse are dreptul să acceseze entitatea autentificată?”*

La ora actuală există o serie de pattern-uri care descriu procesul de autentificare și identificare [109]. Un astfel de exemplu este pattern-ul Automated I&A design alternatives: autentificare pe bază de parole, autentificare folosind metodele biometrice (ex. imaginea degetului, recunoașterea feței [110], geometria mâinii, recunoașterea iris-ului [111], recunoașterea vocii, etc), hardware token (ex. carduri RFID [112], carduri magnetice, carduri inteligente, carduri de genul SecureID), etc.

Diferitele metode de autentificare pot fi combinate astfel încât să se obțină un nivel al siguranței crescut. Pattern-ul care descrie acest model este „Automated I&A design alternatives”, descriind practic diferite tehnici de autentificare ce pot fi combinate. Conform acestui patern se pot combina diferite metode pentru a avea în cadrul procesului de autentificare și identificare următoarele elemente [113]:

- **Ceva ce știm** – ex. parolă alfanumerică, parolă grafică, TAN (parolă ce funcționează o singură dată). Parolele alfanumerice reprezintă una din cele mai folosite metode de autentificare în rețelele de calculatoare, fiind folosite încă din 1960 [114].

- **Ceva ce avem** – ex. diferite carduri de identificare.

- **Ceva ce suntem** – ex. metode de autentificare biometrice.
- **Unde suntem** – ex. poziția GPS.

Cu cât se combină mai multe componente cu atât crește și siguranța sistemului realizat. Nivelul cel mai scăzut al autentificării se realizează cu componente de genul „ceva ce știm" și crește cu cât adăugăm mai multe componente, până la prezența componentelor de genul „unde suntem", Fig. 5.1 (grafic realizat pe baza [4], [10], [113]).

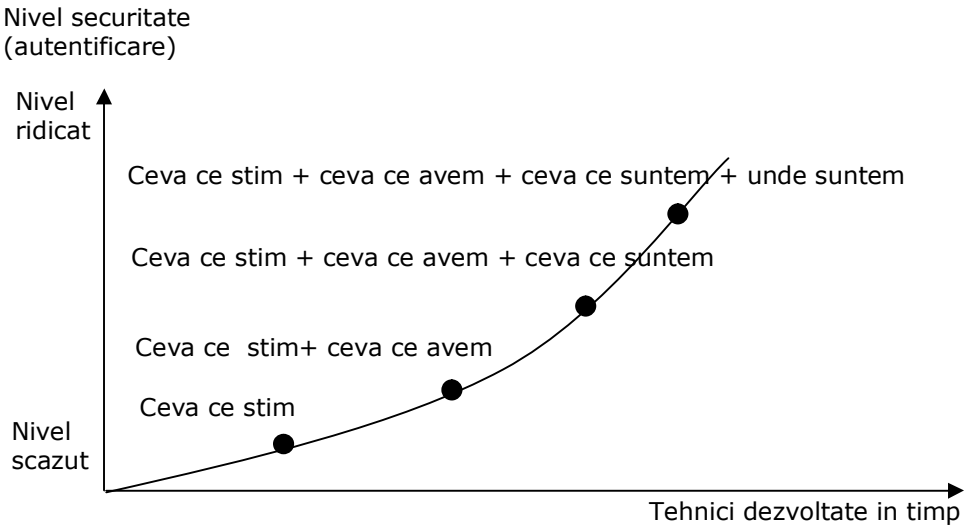


Fig. 5.1 Combinare diferite metode de autentificare – nivel de securitate rezultat

Aceste metode au fost dezvoltate în timp corespunzător necesităților de securitate din ce în ce mai crescute. Așadar, dacă la început s-a folosit user și o parolă, (folosindu-se apoi metode de criptare din ce în ce mai sofisticate pentru parolele create), acestea nu au mai fost suficinte pentru realizarea sistemelor sigure. Acest lucru a dus la dezvoltarea de noi tehnologii și metode de autentificare.

Metodele de autentificare de genul „*ceva ce știm*" au o serie de dezavantaje, ca de exemplu:

- pot fi relativ ușor obținute de către eventualii atacatori. Eventuale metode de determinare a parolelor sunt: metoda dicționarului, physing, trimiterea de E-mail-uri în care se cere unei entități, din diferite motive, datele de autentificare, folosirea unor programe spion care pot citi tastatura celui atacat obținând astfel datele necesare, etc.
- nu oferă nivel ridicat de securitate, dar în schimb este avantajoasă din punct de vedere al costului aferent. În același timp, prin îmbinarea acestei metode cu alte metode se poate obține un sistem ce îndeplinește cerințele dorite. Un

astfel de exemplu este combinarea dintre Secure ID împreună cu o parolă fixă, utilizat adesea pentru accesarea rețelelor VPN;

Metodele de genul „*ceva ce avem*" au ca și principal dezavantaj faptul că pot fi relativ ușor falsificate.

Metodele de genul „*ceva ce suntem*" au ca și principale:

- **Dezavantaje**: costuri mari, rate de eroare de multe ori ridicate (depinzând în mod direct de metoda biometrică aleasă).

- **Avantaje**: oferă un grad ridicat de siguranță (fiecare purtând cu sine ID-ul cu care se face autentificarea) fiind totodată ușor de folosit. Și asupra acestora se pot realiza diverse tipuri de atacuri, dar acestea sunt mai dificil de realizat decât pentru o simplă parolă sau card de identificare. Tipurile de atacuri și ușurința cu care se pot realiza depind în mod direct de tipul de autentificare biometrică folosit.

În cazul proiectului realizat s-au folosit carduri de identificare RFID („ceva ce avem") pentru identificarea subiecților. Dezvoltări pentru viitor vor trebui realizate în acestă zonă prin îmbinarea acestei metode cu o metodă dintr-un nivel mai înalt de securitate precum și colaborări cu firme ce oferă module de autentificare specializate. Astfel, se vor putea realiza inclusiv adaptări (customizing) pentru diverși clienți în funcție de necesitățile fiecăruia.

5.2 Autentificare prin intermediul RFID

Cardurile RFID fac parte din metoda de autentificare hardware token. Tehnologia RFID care stă la baza cardurilor RFID folosește:

- **Un tag** (transponder: transmitter - receiver), care poate fi atașat de un sistem ce se dorește a fi urmărit sau identificat: produs, card RFID, acte de identificare (pașapoarte), etc. De asemenea, astfel de tag-uri pot fi inserate inclusiv în corpul uman sau la animale. Tag-urile pot fi: read only, write once, read-write, o altă clasificare a lor fiind în Tag-uri pasive (fără baterie sau sursă de alimentare), semi-pasive (folosește sursă de alimentare pentru logica din chip, dar nu transmit date la cititor doar dacă este interogat) și active (cu baterie, folosite de exemplu pentru comunicarea cu GPS).

- **Citiror sau/inscriptor** (transceiver) al tag-urilor RFID.

- **Unitate de procesare** a datelor cu un software adecvat care preia datele de la cititorul RFID, le prelucrează și face conexiunea cu restul informațiilor atașate ID-ului tag-ului RFID, informații păstrate într-o bază de date.

Pentru definirea proprietăților unui sistem RFID trebuie luate în considerare trei elemente de bază: sursa de alimentare (tag-uri pasive, active, seimi-active), frecvența și memoria de care dispun, Fig. 5.2.

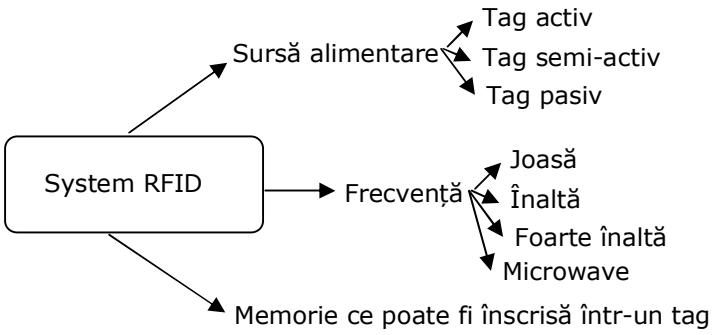


Fig. 5.2 Elemente de bază pentru definirea proprietăților sistemelor RFID

O clasificare a tipurilor de frecvențe, a razelor de acțiune corespunzătoare, precum și distanța la care pot fi citite poate fi găsită [115].

Dintre standardurile implicate în definirea tehnologiei RFID se pot aminti: International Organisation of Standardisation (ISO), EPCglobal Inc și European Telecommunications Standards Institute (ETSI). Pentru Europa ETSI EN 302 208 a stabilit pentru RFID frecvențe împărțite în 15 canale [116].

Structura sub formă de schemă bloc a sistemului RFID pentru cazul proiectului realizat este compusă din elementele prezentate în Fig. 5.3.

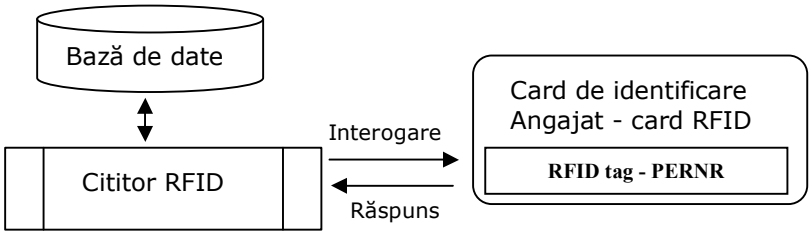


Fig. 5.3 Structura RFID pentru implementarea QBAC

Tag-urile RFID ale subiecților ce se autentifică la unul din obiectele protejate vor conține ID-urile aferente acestora (PERNR) din HR. În momentul în care un angajat introduce cardul de identificare în cititorul RFID se va citit tag-ul (ID-ul corespunzător). Cardurile alese sunt carduri standard, nedispunând de mecanisme de criptare al datelor, carduri ce pot fi inscripționate o singură dată (având tag-uri pasive - cele mai folosite și mai ieftine tipuri de tag-uri). Astfel, informația de pe aceste carduri va putea fi citită doar atunci când sunt interogate de către cititorul RFID, nedispunând de propria sursă de alimentare. După citirea acestora se poate determina angajatul care dorește să realizeze o operație de logare sau delogare în/din sistem. Restul datelor angajatului respectiv sunt păstrate într-o bază de date pe server. Cardurile de identificare RFID folosite nu au fost tipărite în exterior cu informații de genul: poză, nume, prenume, realizarea acestei operații necesitând o imprimantă specială. Pentru dezvoltări viitoare se poate lua în considerare

dezvoltarea unui software pentru tipărirea acestor date, folosind anumite mostre de format predefinite și folosind informațiile stocate în modulul de resurse umane.

Tehnologia RFID dispune de o serie de avantaje, dar și numeroase dezavantaje, cum ar fi diversele atacuri (ex. clonare neautorizată, urmărire neautorizată), ce pot fi realizate asupra unei astfel de tehnologii.

În ceea ce privește problemele de securitate ale protocoalelor de autentificare RFID, câteva dintre acestea pot fi găsite [117].

Dintre premisele alegerii acestei tehnologii pentru autentificarea angajaților se pot menționa:

- **Tehnologie de viitor** ce se dezvoltă din ce în ce mai mult. Această tehnologie este dezvoltată în primul rând din dorința de înlocuire a codurilor de bare cu tag-uri RFID, ajutând în crearea sistemelor moderne de inventariere, urmărire a produselor, crearea profilelor cumpărătorilor, etc.

- **Ușurință în extinderea funcționalităților –** cu ajutorul acestei tehnologii se poate extinde funcționalitățile oferite. Astfel, se poate extinde QBAC cu efectuarea automată a pontajului la ieșirea și intrarea angajaților din/în firmă, cu accesul limitat la anumite zone din firmă, accesul în parcarea firmei, etc.

- **Permite folosirea de chip-uri implantate** – dacă la început au fost create chip-uri ce au fost implantate în corpul animalelor pentru realizarea anumitor verificări (ex. vaccinuri realizate, firma aparținătoare, boli), în prezent există din ce în ce mai multe persoane ce au implantate chip-uri cu tag-uri RFID. Ca și exemple reale ale folosirii acestora în corpul uman, se pot aminti: sistemul medical (anumite state în USA), folosirea acestora ca și metodă de autentificare în cluburi (ex. Spania, Franța). De exemplu, în sistemul medical, tag-ul RFID conține un cod care citit și interconectat cu o bază de date medicală oferă toate informațiile legate de identitatea celui care are implantat chip-ul, bolile de care acesta a suferit, tratamentele parcurse, grupă sanguină, etc. Așadar migrarea de la carduri RFID la chipuri RFID în cadrul proiectului de față ar fi una ușoară.

- **Folosirea tag-urilor și a bazelor de date asociate** oferă avantajul că subiecții proiectului nostru au structura ce se potrivește acestui model. Baza de date conține toate informațiile asociate tag-urilor, iar cheia de identificare este un număr ce reprezintă identitatea subiecților în cadrul firmei.

Dintre dezavantajele acestei tehnologii se pot aminti:

- pot fi folosite pentru spionarea persoanelor, mai ales în combinare cu GPS;

- posibilitate de clonare a tag-urilor RFID;

- nivelul de standardizare și politicile de securitate, criptare nu sunt încă foarte bine definite;

- neâncredere în rândul populației.

Folosirea tehnologiei RFID este controversată în prezent, organizațiile ce se ocupă cu drepturile omului fiind în alertă avându-se în vedere că această tehnologie poate să se transforme într-o metodă prin care sfera privată să fie încălcată, iar persoanele vor putea fi urmarite, obținându-se astfel informații despre produsele achiziționate, date personale, date medicale, date din E-pass, etc.

Conform [118], [119] trebuie să se respecte anumite drepturi ale persoanelor care folosesc sisteme ce conțin tehnologiei RFID. De exemplu:

- dreptul de a fi informat dacă un produs achiziționat folosește tehnologia RFID;
- dreptul de a înlătura sau distruge eventualele tag-uri RFID la cumpărarea produselor respective;
- pe baza acestei tehnologii să nu se piardă alte drepturi (ex. dreptul de a returna un produs);
- dreptul de a fi informat asupra informațiilor ce se stochează într-un tag și dreptul de a fi informat despre datele ce se stochează în baza de date asociată;
- dreptu de a ști cine, unde și de ce citește tag-ul RFID.

Există diverse metode de securitate ce protejează datele din tag-urile RFID astfel încât acestea să nu poată fi citite de către oricine, sau să fie greu clonabile. Câteva dintre aceste metode pot fi găsite [120].

Pentru comunicarea directă cu cititoarele RFID, SAP pune la dispoziție clienților săi componenta SAP-ALL [121]. Există totodată diverse firme care oferă aplicații soft de legătură dintre platforma SAP NetWeaver și PLC-uri (ce au un modul de interfață cu cititorul RFID). Componenta SAP-All este folositoare în cazul comunicării directe cu cititoarele RFID ajutând în procesul de logistică, urmărire al produselor, etc. Deoarece, pentru realizarea obiectivelor proiectului de față trebuie să se comunice cu cititorul RFID prin intermediul unui PLC, folosirea SAP-ALL nu este adecvată. O soluție a problemei ar fi fost folosirea de aplicații soft livrat de firme terțe (ex. PEAK Automation Controller – Enterprise 5.0). Această soluție este însă foarte costisitoare deoarece consultanții SAP ai firmei respective trebuie să facă o analiză a necesităților proiectului și apoi să implementeze soluția lor. Aceste firme au de obicei dezvoltată o aplicație soft complexă cu o multitudine de drivere, astfel încât să ofere comunicare nu doar cu PLC (ale diverselor firme) ci și cu alte dispozitive. Nu am gasit însă nici un produs de firmă, care să ofere o variantă de test în vederea implementării și testării.

Din aceste considerente, am dezvoltat propria soluție de comunicare între PLC și platforma SAP NetWeaver, ajutându-ne de tehnologia OPC (OPC server și OPC client). Aceste produse soft se oferă în variantă de test (gratuit), reprezentând tehnologia standard pentru stabilirea comunicării între o stație de lucru și unul sau mai multe PLC-uri. Necesitățile proiectului de față se referă la PLC-uri ale unei

singure firme (Siemens), acest lucru simplificând problema și nejustificând costul foarte ridicat al folosirii unor produse adiționale.

Concluzii:

În cadrul acestui capitol:

- s-a realizat o analiză a diferitelor metode de autentificare existente, precum și a modului în care acestea pot fi îmbinate în vederea creșterii nivelului de securitate al aplicației realizate;
- s-a prezentat modul de folosire al tehnologiei RFID pentru autentificarea subiecților, justificând motivele alegerii acestei tehnologii;
- s-au analizat drepturile ce trebuiesc respectate subiecților pentru care se acordă autentificare pe baza unei astfel de tehnologii.

Concluzionând, se poate spune că metoda de autentificare folosită oferă un nivel scăzut al securității, mai ales dacă nu se folosesc protocoale speciale de securizare astfel încât cardurile RFID să fie greu clonate și citite de către oricine interoghează respectivul card. Această metodă de autentificare poate fi schimbată cu o altă metodă cu un grad de siguranță mai ridicat sau combinată cu alte tipuri de metode de autentificare. Acest lucru nu va schimba logica QBAC deoarece metoda de de autentificare nu face parte din calculul dreptului de acces bazat pe calificative dezvoltată, necesitând doar pentru autentificarea subiecților la nivelul obiectelor protejate.

Pentru a se putea realiza adaptări în funcție de necesitatea fiecărui client și pentru a crește nivelul de securitate al sistemului realizat, se va colabora cu firme specializate (ex. www.pcs.com) în autentificare. Astfel, se vor integra în logica QBAC diferite module de autentificare oferite de acceste firme, modificările care trebuiesc realizate în acest caz referindu-se la integrarea respectivei metode de autentificare (ex. bază de date pentru stocarea datelor biometrice, hardware suplimentar) în Framework-ul QBAC.

Pentru comunicarea directă dintre un cititor RFID și platforma SAP NetWeaver se poate folosi componenta standard SAP-ALL. Datorită faptului că, în cazul de față comunicarea se realizează prin intermediul unui PLC, folosirea acestei componente nu este adecvată.

6. Aplicația realizată folosind OPC server, OPC client și Step7

În cadrul capitolului de față se prezintă soluțiile propuse în vederea comunicării dintre platforma SAP netWeaver și PLC. Totodată, se prezintă modul în care s-a folosit OPC server și OPC client pentru înscrierea datelor în PLC, precum și pentru citirea datelor din PLC, precum și modul în care s-a programat PLC-ul, folosind aplicația soft Step7. În acest context, se vor prezenta funcționalitățile de bază ale OPC precum și modul de utilizare al limbajelor de programare Step7 pentru implementarea metodei de acces dezvoltate.

6.1 Programare PLC-ului Simatic S7-300 utilizat

Pentru programarea PLC-ului Siemens s-a utilizat Step7. Folosind această aplicație soft pentru programarea PLC-ului, se dispune de o serie de avantaje, acestea fiind de un real ajutor în procesul implementării. Dintre acestea amintim [122], [123], [124]:

- ușurintă în configurare și parametrizare hardware;
- ușurință în configurarea legăturii cu PLC-ul și a proprietăților necesare comunicării;
- pune la dispoziție diverse limbaje de programare și oferă posibilitatea extinderii funcționalităților Step7 cu unelte adiționale;
- ajustare module adrese;
- testare programe realizate atât în varianta simulată cât și în varianta online necesară vizualizării stării variabilelor în timpul rulării programelor în PLC;
- posibilitate de modularizare a programelor realizate prin împărțirea în diverse tipuri de blocuri;
- posibilitate de îmbinare a diverselor tipuri de limbaje de programare, fiecare bloc putând fi programat într-un limbaj care se potrivește cel mai bine nevoilor respective.

Aplicația soft Step7 oferă principial trei limbaje de programare (LAD, FBD, STL) care pot fi extinse cu limbajele aditionale (ex. S7-SCL, S7-PLCSIM, S7-Graph). În cazul proiectului de fata s-a folosit programarea S7-SCL îmbinată cu FBD, iar

pentru simularea anumitor funcționalități înaintea download-ării programelor în PLC s-a folosit S7-PLCSIM.

S-a ales folosirea S7-SCL pentru implementarea algoritmilor necesari decodării dreptului de acces, pentru determinarea adreselor fizice la care se acordă drept de acces, precum și pentru determinarea adreselor ieșirilor ce vor fi comandate. Astfel, s-au putut îmbina blocurile create cu acest limbaj, cu blocurile create cu ajutorul unui alt limbaj suportat de catre software-ul Step 7. Programul S7-SCL s-a realizat în editorul de text pus la dispozitie de S7-SCL iar, dupa compilare, s-au generat blocurile Step 7 corespunzatoare (Fig. 6.1).

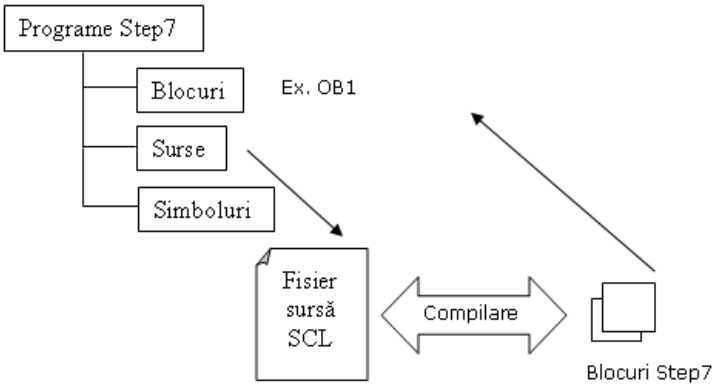


Fig. 6.1 Generarea blocurilor Step7 in urma compilarii programului S7-SCL realizat

Blocul cel mai important este blocul OB1, acesta reprezentând programul principal care se execută ciclic si care s-a implementat folosind S7-SCL. Din cadrul acestuia s-au apelat apoi în funcție de necesități celalte blocuri create.

Pentru definirea variabilelor globale, prin intermediul cărora să se poata citi și înscrie date din/în PLC folosind OPC server, OPC client s-au folosit blocurile de tipul DB (Data Block).

Limbajul FBD s-a folosit pentru restul operațiilor ce nu necesita codare, beneficiand de avantajul că disune de un număr mare de funcții ce pot fi folosite prin funcționalitate de dragg & dropp. Structura proiectului rezultat este prezentată în **Anexa A9.**

Pentru un PLC din familia Siemens, intrările și ieșirile sunt împărțite în grupuri de 8 intrări sau ieșiri:

- Ix.y, unde I reprezinta tipul de adresă de tip input, x reprezintă byte-ul de adresă, iar y bit-ul de adresă.

- Qz.k, unde Q reprezinta tipul de adresă output, z byte de adresă, k bit-ul de adresă.

Pentru a nu se lucra cu adresele fizice ale intrărilor și ieșirilor, s-au realizat aliace-uri pentru acestea, urmând ca aceste aliace-uri sa fie importante în OPC server.

Configurația hardware și programele împărțite în diverse tipuri de blocuri și realizate folosind diversele limbaje de programare s-au download-at de pe stația de lucru în PLC folosind conexiunea MPI (Fig. 6.2).

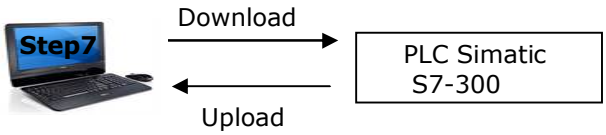


Fig. 6.2 Download și Upload al programelor de la stația de lucru la/de la PLC

6.2 Soluțiile dezvoltate pentru comunicarea dintre platforma SAP NetWeaver și PLC

Dupa cum s-a menționat, pentru realizarea comunicării dintre platforma SAP NetWeaver și PLC se pot folosi soluții oferite de diverse firme (ex. PEAK Automation Controller Enterprise 5.0). Aceste soluții sunt costisitoare, și nu se oferă aplicațiile soft sub formă de versiune trial, care să poata fi testate. Acesta este motivul pentru care s-au dezvoltat propriile soluții folosind tehnologia standard OPC.

Un OPC server este o aplicație soft standardizată ce ajută în comunicarea cu diferite elemente de proces. Necesitatea folosirii OPC server și OPC client a apărut la începutul anilor 1980 [126], dar definirea acestui standard a fost un proces ce s-a întins pe mai mulți ani, primele specificații OPC fiind definitivate în anul 1996. Fundația care se ocupă de standardul OPC este OPC fundation (http://www.opcfoundation.org/). Aceasta asigurând că aplicațiile create de diverși producători se pot combina în cazul în care aceste aplicații soft sunt certificate [127].

OPC server și OPC client a devenit un standard oferind posibilitatea de a schimba informații între diferite componente ce necesită date sau necesită înscrierea de date într-un proces industrial.

Pentru scopul proiectului de față, această tehnologie standardizată va fi folositoare pentru schimbarea de informații între platforma SAP NetWeaver și PLC-ul folosit, Fig. 6.3.

Ca și logică de implementare OPC folosește varianta client – server unde server-ul este o aplicație soft ce interacționează cu hardware-ul, iar client-ul este oferit sub forma unui ActiveX, ce permite citirea și scrierea de date în/din OPC server. Comunicarea dintre OPC server și hardware se face prin intermediul așa numitelor TAG-uri definite de către programator în funcție de valorile necesare. În cazul metodei de acces dezvoltate s-au importat aliace-urile definite în PLC. Aplicațiile soft OPC server și OPC client poate fi folosit pe un singur calculator (cazul de față), sau pe mai multe calculatoare.

Așadar, pentru proiectul de față se propun două **modalități de comunicare** cu platforma SAP NetWeaver:

- Folosind **Web Service** – în acest caz s-a creat o aplicație Visual Basic care să integreze ActiveX-ul de comunicare cu OPC server și s-a folosit Web Service-ul (creat în cadrul platformei SAP) pentru a obține date de la

platforma SAP NetWeaver. Folosind versiunea cu Web Service se poate comunica și direct cu PLC-ul în cazul în cazul în care se folosesc module adecvate.

- Folosind **OPC** client sub forma ActiveX direct în limbajul ABAP având ca și avantaj faptul că, comunicarea cu OPC server se face direct.

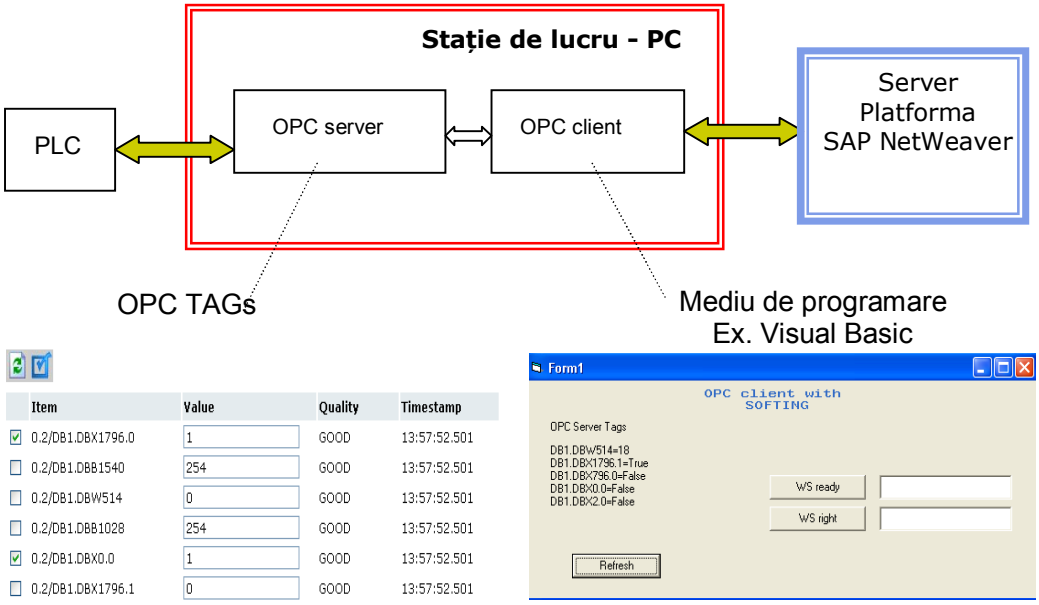


Fig. 6.3 Comunicare platformă SAP NetWeaver – PLC folosind OPC server, OPC client

Există foarte multe firme care oferă software OPC server și OPC client pentru comunicarea cu un PLC din familia Siemens. La referința [128] se pot găsi o serie de aplicații soft OPC care pot fi folosite în variantă de test. Pentru implementarea QBAC s-au ales aplicațiile soft dezvoltate de firma SOFTING, atât ca și OPC server cât și ca OPC client. OPC server s-a instalat pe stația de lucru, ce comunică direct cu PLC-ul iar OPC client (în variantă ActiveX) a fost folosit pentru a comunica cu OPC server din mediul de programare.

Dintre avantajele folosirii OPC se pot aminti:

- produsele mai multor firme pot fi combinate;
- este o tehnologie open standardizată;
- poate fi folosită în orice combinație OPC server, OPC client;
- accesarea unui număr mare de clienți;
- posibilitate de a avea interfață browser.

Concluzii:

În cadrul acestui capitol:

- s-a prezentat modul de folosire a tehnologiei OPC (variantă server și client) pentru înscrierea datelor în PLC și citirea datelor din acesta de la nivelul unei aplicații;
- s-au propus două metode de comunicare dintre platforma SAP NetWeaver și PLC;
- s-a implementat algoritmul necesar logicii QBAC la nivel PLC, prezentând totodată uneltele software folosite în procesul de programare și configurare hardware.

Concluzionând, se poate spune că, pentru comunicarea dintre platforma SAP NetWeaver și PLC se pot alege mai multe soluții. Există firme specializate care oferă aplicații soft pentru o astfel de legătură, dar aceste soluții sunt foarte costisitoare și nu se oferă versiuni trial, care să poată fi testate. Acesta a fost motivul pentru care am propus două soluții, ajutându-ne în acest scop de tehnologia standard OPC.

Prima soluție folosește un Web Service pentru obținerea dreptului de acces de la platforma SAP NetWeaver, iar apoi într-un limbaj de programare (ex. Visual Basic, C) se utilizează OPC client pentru înscrierea datelor respective în PLC. Acesată soluție poate fi folistă și în cazul în care anumite obiecte protejate sunt localizate în afara companiei dar, are dezavantajul că obținerea datelor de la server poate să fie blocată în cazul în care serverul este ocupat.

Ca urmare al oferirii dreptului de acces sub forma unui Web Service se poate folosi și o variantă de comunicare directă cu PLC-ul, fără a folosi tehnologia OPC.

A doua soluție este folosirea OPC (sub forma Active X) direct în limbajul ABAP, comunicarea în acest mod făcându-se mai rapid decât varianta cu Visual Basic.

7. STRUCTURA STANDULUI UTILIZAT PENTRU TESTAREA PATTERN-ULUI QBAC, REZULTATE EXPERIMENTALE

Capitolul de față este structurat pe doua parți. În prima parte se face o succintă trecere în revistă a avantajelor folosirii PLC-urilor, urmând apoi să se prezinte structura PLC-ului utilizat.

Partea a doua este dedicată prezentării standului utilizat pentru testarea metodei de acces propusă, prezentării rezultatelor experimentale precum și analizării sistemului distribuit obținut.

7.1 Structura PLC-ului utilizat

PLC-urile sunt folosite în sistemele industriale pentru controlul diverselor procese.

Dintre avantajele folosirii PLC-urilor amintim [129], [130], [131]:

- ușurință în instalare, configurare și programare;
- oferă diverse module care pot fi adăugate în funcție de necesități;
- oferă posibilitatea conectării unui număr larg de sisteme;
- posibilitate de adăugare a mai multor unități de procesare de genul CP;
- diverse funcționalități de diagnostigare care ajută să se obțină rapid informații privind erorile ce apar;
- se pot comanda sisteme ce dispun de un număr foarte mare de intrări și ieșiri;
- oferă posibilitatea operațiilor de download-are a programelor realizate și de upload-are din memoria PLC la nivelul calculatorului;
- oferă posibilități standardizate (OPC server, OPC client) de comunicare cu un calculator pentru citirea și înscrierea datelor din/în acesta;
- costuri relativ scăzute, depinzând de firmă și tip de module folosite.

Acestea au fost și motivele alegerii controller-elor de tip PLC pentru realizarea proiectului.

Pentru crearea comenzilor către obiectele protejate, precum și pentru preluarea datelor de la acestea și de la cititoarele RFID s-a ales să se utilizaze un PLC de tipul SIMATIC S7-300, produs de firma Siemens. Acest tip de controller are o structură modularizată și conține următoarele componente principale:

- sursă de alimentare (PS);
- CPU pentru stocarea și procesarea programelor realizate;
- rack pentru interconectarea diverselor module folosite.

Structura PLC-ului SIMATIC S7-300 utilizat pentru testarea metodei de acces controlat dezvoltate în teză dispune de urmatoarele componente: sursă de alimentare (DC24V), modul de intrare (SM321: DI 16XDC24V), modul de ieșire (SM322: DO 16X24V/0.5A) și interfață pentru cititorul de carduri RFID (CP 341-RS232C), Fig. 7.1.

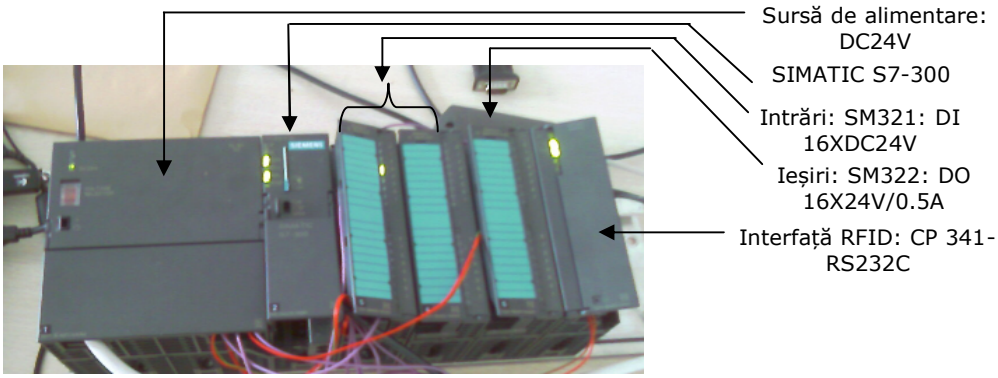


Fig. 7.1 Structura PLC din familia Siemens utilizat

Modulul RFID al PLC-ului s-a conectat prin intermediul unei interfețe seriale (RS232) la cititorul/inscriptorul RFID. Un cititor/inscriptor RFID poate citi date și scrie date. Scrierea cardurilor RFID s-a facut cu ajutorul programului standard oferit de producător. Pentru realizarea acestei operații cititorul RFID s-a conectat direct la portul USB al calculatorului, prin intermediul unui adaptor corespunzător. La nivel PLC s-a folosit acest dispozitiv doar cu funcția de citire, aceasta fiind singura funcționalitate necesară în cadrul proiectului considerat.

7.2 Standul de test realizat

Standul de test necesar testării metodei de acces controlat propusă este format: dintr-un PLC SIMATIC S7-300, o mașină de test, un cititor RFID și un panou de comandă.

Mașina de test (obiectul protejat) utilizată pentru testarea metodei de acces controlat este de tip lego, putand fi construită în mai mult variante dispunând de 3 ieșiri și 3 intrări.

Deoarece intrările sunt implementate în acest caz sub forma senzorilor, subiectul neavând posibilitatea să activeze acele intrări (să apese butoanele respective) s-a creat un panou de comandă. Cu ajutorul acestuia, după efectuarea procesului de autentificare și autorizare, se va oferi drept de accesare doar a butoanelor (intrărilor) pentru care există autorizarea corespunzătoare. Pentru conectarea obiectului protejat la PLC a fost nevoie de o interfață (adaptare tensiuni), interconectarea făcându-se direct la motoare, fără a folosi interfața standard livrată de producător. În acest caz se vor comanda cele 3 motoare (direcție dreapta, directie stânga) ce pun în mișcare obiectul protejat, putând astfel acorda drepturi de acces pentru cele 6 intrări de test rezultate.

În Fig. 7.2 se prezintă structura **standului de test** utilizat.

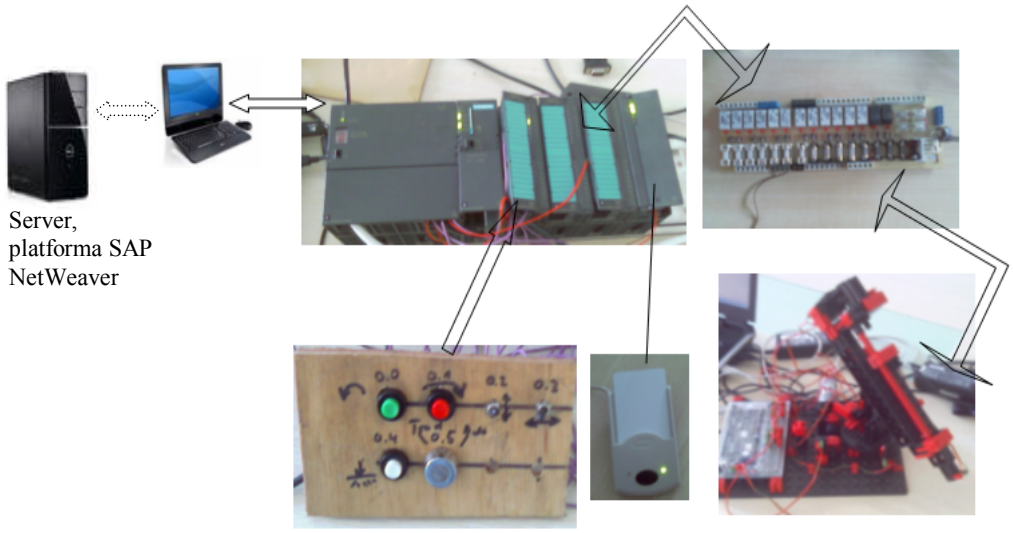


Fig. 7.2 Structura standului de test utilizat

Butoanele corespunzătoare panoului de comandă sunt conectate la intrările PLC-ului, iar ieșirile ce vor comanda obiectul protejat sunt conectate la ieșirile PLC-ului. Cititorul cardurilor RFID este conectat la modulul RFID, așteptând logarea sau delogarea unui angajat. Fiecare mașină de test va trebui să fie dotată cu propriul cititor de carduri RFID, astfel încât să se știe exact la care dintre mașini se dorește logarea/delogarea, respectiv pentru care dintre mașini se determină dreptul de acces al respectivului subiect, sau pentru care mașină se înscriu date în baza de date.

La momentul inițial nici un subiect nu este logat la mașina de test, așadar la apăsarea butoanelor nu se va trimite nici o comandă, toate intrările fiind fără autorizare. După ce un subiect se loghează și primește dreptul de acces codat într-un întreg, se va decoda acel drept, se vor determina adresele fizice la care angajatul are drept de acces și automat adresele fizice ale ieșirilor ce vor fi comandate în cazul activării intrărilor respective. Așadar, în această etapă (sesiune de lucru la nivel PLC) subiectului i se va permite folosirea butoanelor pentru care are drept de acces,

restul butoanelor nu vor genera nici o comandă în cazul apăsării acestora. După ce subiectul respectiv se va deloga toate intrările vor fi fără autorizare.

Pentru comunicarea dintre PLC și stația de lucru (PC) se poate folosi fie adaptorul MPI, fie un alt tip de rețea de genul Ethernet, Wirles, etc. Deoarece pentru testarea metodei de acces se dispune de un singur obiect protejat, folosirea unui modul Ethernet ar fi redundantă. Un singur PLC a fost suficient pentru comunicarea cu obiectul protejat și autentificarea la nivelul obiectului protejat. Din acest motiv pentru realizarea acestei comunicări s-a folosit adaptorul MPI (conexiune USB la PC, adaptor), Fig. 7.3.

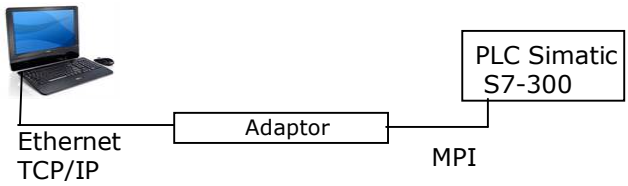


Fig. 7.3 Comunicare stație de lucru - PLC

Mai multe detalii legate de comunicarea PC – PLC folosind Ethernet în vederea conectării mai multor PLC-uri la un calculator se găsesc la indicația bibliografică [132].

7.3 Rezultate experimentale, analizarea sistemului distribuit obținut

Pentru obținerea **rezultatelor experimntale** se vor crea calificative pentru mașina de test, se vor asigna aceste calificative pentru trei angajati, iar apoi se vor simula operații de logare și de delogare ale angajaților la nivelul mașinii de test.

În acest mod se va putea verifica funcționarea metodei de acces bazată pe calificative, dezvoltată în teză.

După cum s-a menționat, pentru mașina de test se poate acorda drept de acces pentru 6 intrări, notate: $I_{0.0}, I_{0.1}, I_{0.2}, I_{03}, I_{04}, I_{0.5}$.

Pentru mașina de test s-au creat trei calificative (Installer, Operator, Tool_Setter) și s-au asignat acestora următoarele intrări:

- Operator: $I_{0.0}, I_{0.2}$;
- Installer: $I_{0.1}, I_{0.4}$;
- Tool_Setter: $I_{0.3}, I_{0.4}$.

În Tabelul 7.1 se prezintă id-urilor angajaților de test, calificativele ce au fost asignate acestora, dreptul de acces rezultat în urma logării, precum și viteza de execuție a obținerii datelor respective la nivelul platformei SAP NetWeaver.

Pentru testarea metodei de acces și în cazul apariției redundanțelor s-au asignat calificative multiple angajaților de test.

Îmbunătățirea vitezei de execuție la nivelul implementării poate fi realizată prin renunțarea la programarea clasică (Function Module) și folosirea componentei SAP XI pentru Web Service. Dezavantajul este însă acela că, în acest caz trebuie să se folosească o extra componentă SAP, ceea ce duce la creșterea considerabilă a prețului aferent software-ului.

Subiect – id, nume	**Calificative**	**Timp de execuție [microsecunde]**	**Drept de acces rezultat**
3, Kadenbach Mathias	Installer, Tool_setter, Operator	67,243	31
2, Stefan Schmidt	Operator, Tool_setter	52,278	29
1, Hans Maier	Installer	42,197	18

Tabelul 7.1 Rezultate experimentale

Rezultatele obținute sunt relevante pentru mediul în care s-a realizat implementarea, neavând caracter absolut.

În continuare se va **analiza sistemul distribuit** obținut din punct de vedere al cerințelor unui stfel de sistem, conform [35], [40], [41].

Un sistem distribuit trebie să:

- fie **transparent**, respectiv să ascundă faptul că resursele sunt distribuite de-a lungul rețelelor și pe mai multe calculatoare (transparent system);
- fie **deschis**, respectiv să fie ușor de configurat, flexibil, ușor de adăugat noi componente, să folosească standarduri și tehnologii create de diverși producători (open system);
- fie **scalabil**, respectiv ușor de manageriat chiar și în cazul în care numărul de resurse crește (scalable system). Internetul este un exemplu de astfel de sistem scalabil. În 1979 erau 188 calculatoare, 1989 erau 130.000 calculatoare iar 1999 erau 56.218.000 calculatoare (conform [133]);
- fie **concurent**, respectiv resursele să poată fi accesate de mai mulți utilizatori simultan (concurent system);
- **asigure securitatea** necesară (secure system).

Pentru a analiza problematica **transparenței sistemului** distribuit din punct de vedere al **transparenței erorilor** s-a simulat apariția unei erori la comunicarea dintre PLC și server. În cazul în care la nivelul aplicației Visual Basic, sau al aplicației ABAP (cazul folosirii Active X) se simulează apariția unei erori prin introducerea unei valori nepermise în PLC, se obțin ca și rezultat o eroare software la PLC, astfel încât întrega execuție ciclică va fi anulată, Tabelul 7.2.

Toate valorile introduse în PLC prin intermediul OPC server OPC client, la fel ca și restul intrărilor folosite sunt verificate dacă îndeplinesc tipul cerut, înainte de a

fi introduse în PLC sau în baza de date. Chiar dacă acest raspuns vine de la server această verificare este indicată a fi facută pentru a înlătura o situație identică cu cea simulată, ce ar duce la generarea unei erori software în PLC-ul în care a înscris date respectivul OPC server.

Valoare introdusă in PLC folosind OPC server, OPC client	**Rezultat, intrări la care se acordă drept de acces**
31	$I_{0.0}, I_{0,1}, I_{0.2}, I_{0,3}, I_{0,4}$
18	$I_{0.1}, I_{0.4}$
- 0.9	Eroare software PLC, valoare necorespunzătoare

Tabelul 7.2 Analiza reacției la inserarea unei eroari în PLC

De asemenea, la nivel de server se verifică datele introduse în baza de date, generându-se mesaje de eroare în cazul în care se dorește introducerea unor valori necorespunzătoare, Tabelul 7.3.

Valoari introdusă în baza de date prin intermediul FM sau WD	**Rezultat**
FM -> PERNR 3, Machine_ID 101.004.RBT	The Machine 101.004.RBT don't exist!
WD-> inserare mașină nouă, cazul în care grupa în care se dorește introducerea noii mașini nu există	Groupe id 003 don't exist! - **Anexa A10**
WD-> importarea în masă a ID-urilor angajaților din HR, în vederea integrării în conceptul de autorizare	The HR personal number don't exist! - **Anexa A10**

Tabelul 7.3 Analiza privind transparența erorilor la nivel de server

Deoarece dispunem de un sistem distribuit cu comunicare punct la punct, la fiecare moment se știe care este procesul ce a generat eroarea în rețeaua de componente.

Principalele situații ce trebuie tratate pentru a obține informații legate de procesul care a generat o eroare sunt urmatoarele:

- clientul nu poate localiza serverul;
- clientul se blochează nemaiputând să își îndeplinească funcțiile;
- serverul se blochează nemaiputând să își îndeplinească funcțiile;
- mesajele transmise între client și server sunt pierdute.

Pentru analizarea evenimentelor ce pot avea loc la server-ul din structura sistemului distribuit (Fig. 3.19), în cazul căderii (crasch) acestuia s-a utilizat o analiza propusă [41]. Astfel, dacă se notează cu M trimiterea mesajului de răspuns,

cu O executarea operațiilor corespunzătoare și cu C o eventuală cădere a serverului se vor obține următoarele trei situații:

a) O->M->C, executare operații, transmitere mesaj, cădere server, Fig. 7.4. Serverul cade după executarea operațiilor și trimiterea mesajelor.

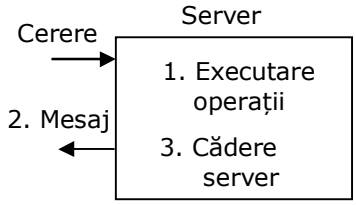


Fig. 7.4 Posibile scenarii de cădere (crasch) al unui server în comunicarea client – server, varianta O->M->C

b) O->C[->M], executare operații, cădere server fără a se mai putea trimite mesaje, Fig. 7.5

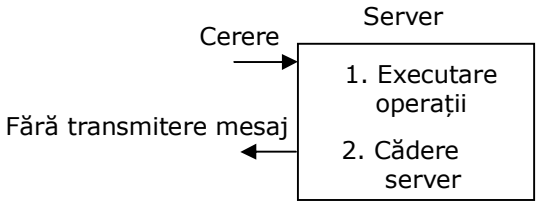


Fig. 7.5 Posibile scenarii de cădere (crasch) al unui server în comunicarea client – server, varianta O->C[->M]

c) C[->O->M], cădere server fără a se mai executa oerațiile necesare și fără transmiterea mesajului, Fig. 7.6

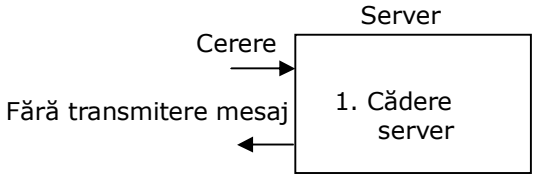


Fig. 7.6 Posibile scenarii de cădere (crasch) al unui server în comunicarea client – server, varianta C->[O->M]

În cazul în care mesajul de răspuns nu mai este trimis clientul nu știe dacă serverul a căzut înainte de realizarea operațiilor sau după realizarea acestora. Stația de lucru (clientul) are însă nevoie de datele de la server (raspunsul acestuia). Tratarea erorilor în cazul situațiilor b) și c) este realizată prin retransmiterea cererii și analizarea rezultatului. În cazul în care se retransmite cererea și după analizarea rezultatului avem situația c) s-au obținut de la server datele necesare, sau pentru o operație de delogare s-a înregistrat în baza de date activitatea și s-a deblocat subiectul și obiectul protejat. În cazul în care a apărut situația b) va trebui să se mai

retrimită odată cererea pentru obținerea rezultatului dorit. Cererea dublă în acest caz este necesară datorită logicii proprii QBAC prin care după o operație de logare se obține dreptul de acces dar se blochează subiectul și obiectul protejat. Printr-o operație dublă se evită situația în care datele nu ar fi obținute de la server, dar subiectul și obiectul protejat ar fi blocate.

Astfel, prin trasmiterea mesajelor și așteptarea feedback-urilor se va ști dacă răspunsul a fost obținut sau nu. De exemplu, la nivelul PLC pentru a ști dacă datele necesare au fost înscrise de către OPC server în PLC s-a folosit un mesaj de răspuns înscris într-o variabilă globală, ce semnalizează că acele valori au fost înscrise, putându-se trece la procesul următor –decodarea dreptului de acces.

Așadar, din punct de vedere al tranparenței erorilor s-a obținut un sistem distribuit ce îndeplinește condițiile necesare, informând utilizatorul asupra apariției anumitor erori (nelăsând-ul să aștepte un răspuns care nu apare din caua unei eventuale căderi ale unui server sau client) și încercând să împiedice apariția unor erori prin verificări efectuate asupra intrărilor.

Modelul de referință ISO al Open Distributed Processing (RM-ODP) [134] a identificat opt tipuri de transparențe, care pot fi discutate la un sistem distribuit. Din punct de vedere al asigurării unei trasparențe totale aceasta este dificil de realizat, necesitând o serie de eforturi adiționale (ex. realizarea anumitor elemente redundante – information redundancy, time redundancy, physical redundacy) și de cele mai multe ori are ca și efect scăderea eficienței. Dezvoltări viitoare vor lua în considerare analize și implementări pentru acest nivel.

Pentru a analiza problematica **securității sistemului distribuit** obținut, trebuie luat în considerare faptul că folosirea unei metode de autorizare sau a unei metode de autentificare cu grad ridicat de siguranță nu duce automat la obținerea unui sistem sigur. Pentru a obține un sistem distribuit sigur trebuie ca pe lângă aceste metode să se realizeze și o programare corespunzătoare, care să înlăture eventualele vulnerabilități, dar să se folosească și alte metode adiționale ca de exemplu algoritmi criptografici.

Dacă se analizează securitatea sistemului distribuit rezulat din punct de vedere al conexiunilor realizate în cazul folosirii Web Service, se va constata că acestuia trebuie să i se acorde un nivel de securitate adecvat. În cazul de față, pentru securizarea acestuia, s-a folosit o simplă parolă și user, acestea fiind însă vulnerabile la atacuri. O parolă și un user sunt ușor de obținut, iar acest lucru ar duce la posibilitatea unui atacator de a înscrie date la nivel server și de a obține datele corespunzătoare de la server. Dreptul de acces este codat, iar logica aflată în spatele acestei codări este specifică aplicației, nefiind posibilă determinarea intrărilor pentru care un subiect are drept de acces. Dar în acest mod se poate determina nu doar dreptul de acces, ci chiar informații de genul id-ul unui subiect, nume, prenume. De asemenea se pot înscrie date false în baza de date.

În ceea ce privește programarea la nivelul serverului s-au implementat elemente de siguranță prin operații de genul:

- verificarea autorizării pentru înscrierea datelor în baza de date;
- verificarea autorizării administratorilor, folosind conceptele de autorizare standard SAP, metodă de acces RBAC;

- prin folosirea componentei SAP ERP HCM ne-am asigurat că se dispune de conceptul SAP de autorizare pentru vehicularea datelor respective;
- înlăturarea eventualelor back-doors prin nefolosirea hard codding;
- înlăturarea eventualelor atacuri de gen SQL injection prin filtrarea intrărilor.

Pentru asigurarea consistenței datelor înscrise în baza de date, SAP pune la dispoziție SAP LUW.

Prin folosirea cardurilor de autentificare RFID se asigură subiecților un mijloc de autentificare modern, ce oferă ușurință în extinderea funcționalităților, dar cu un grad de securitate scăzut în cazul nefolosirii protocolaelor speciale de securizare.

Sistemul distribuit obținut trebuie dezvoltat în versiunile următoare, în special adăugând semnăturile digitale (dacă se folosește Web Service), adăugând protocoale de securizare RFID (în cazul păstrării acestei metode de autentificare) sau eventuale tranziții la chipuri. Totodată colaborarea cu firme specializate în autentificare în vederea integrarii în logica QBAC a diverselor module pentru extinderea tipurilor de autentificare și a securității la acest nivel este de un real ajutor.

Pentru analizarea deschiderii sistemului distribuit obținut se va studia această problemă din punctul de vedere al folosirii tehnologiilor și standarde create de diverși producători. Pentru scopul acestui proiect, la nivelul aplicației soft, s-au folosit o largă gamă de tehnologii și standarde, Tabelul 7.4 prezentând cele mai importante dintre acestea.

Tehnologii și standarde folosite	**Producător sau Organizație**
Web Dynpro ABAP	SAP
Limbajul ABAP	SAP
SAP Netweaver Portal	SAP
SAP ERP HCM	SAP
OPC server	Softig
OPC client	Softig
Visual Basic	Microsoft
Step7	Siemens
Web Serice	Organizația: Web Services Interoperability (WS-I)

Tabelul 7.4 Principalele tehnologii și standarde folosite

Astfel, s-au îmbinat produse de la diverși producători, în funcție de necesități și avantajele oferite, obținând din acest punct de vedere un sistem deschis.

În ceea ce privește **scalabilitatea sistemului**, acesta nu poate fi analizată pentru întregul sistem distribuit deoarece se dispune de o singură mașină de test. În vederea obținerii unui sistem scalabil s-au luat anumite măsuri ce au fost prezentate

în faza de dezvoltare, ca de exemplu împărțirea sistemului pe niveluri. Astfel apariția unei noi intrări la o mașină nu propagă modificări până la nivelele superioare, aceste modificări fiind ușor întegrate la nivelul cel mai de jos de către administratorul respectiv. În acest mod toți subiecții ce dispun de un calificativ la care s-au adăugat noi intrări vor putea automat deservi noile intrări, fără a fi nevoiți să participe la școlarizare, sau să realizeze alte operații.

Analizarea scalabilității sistemului distribuit rezultat precum și a concurenței sistemului vor putea fi realizate doar pentru variante ce se vor dezvolta în viitor.

Concluzii:

În cadrul acestui capitol:

- s-a prezentat PLC-ul Siemens folosit pentru testarea metodei de acces dezvoltate în prezenta teză;
- s-a prezentat standul de test utilizat, stand compus dintr-un PLC SIMATIC S7-300, o mașină de test, un cititor RFID și un panou de comandă;
- s-a testat funcționarea metodei de acces simulând scenarii de logare și delogare;
- s-a analizat sistemul distribuit din punct de vere al funcționalităților ce trebuiesc îndeplinite de către un astfel de sistem.

Concluzionând, se poate spune că metoda de acces dezvoltată funcționează corespunzător specificațiilor, subiecții putând astfel să se logheze și delogheze obținând dreptul de acces de la server (platforma SAP NetWeaver) corespunzător calificativelor care le-au fost asignate de către admnistrator.

În urma testelor realizate se poate spune că sistemul distribuit este transparent din punct de vedere al transparenței erorilor, este un sistem deschis, un sistem ce necesită dezvoltări ulterioare pentru asigurarea securității. Scalabilitatea și concurența nu au putut fi testate deoarece dispunem de o singură mașină de test.

În vederea obținerii unui sistem scalabil s-au luat anumite măsuri ce au fost prezentate în faza de dezvoltare, de exemplu împărțirea sistemului pe niveluri și realizarea codării. Astfel, apariția unei noi intrări la o mașină nu propagă modificări până la nivelele superioare, aceste modificări fiind ușor întegrate la nivelul cel mai de jos de către administratorul respectiv. În acest mod toți subiecții ce dispun de un calificativ la care s-au adăugat noi intrări vor putea automat deservi noile intrări, fără a fi nevoiți să participe la școlarizare, sau să realizeze alte operații. Totodată folosirea codării pentru transmiterea datelor între platforma SAP NetWeaver și PLC duce la minimizarea efortului necesar implementării schimbărilor aduse la nivel de obiect protejat.

8. CONCLUZII, CONTRIBUȚII ADUSE ȘI DEZVOLTĂRI PENTRU VIITOR

În cadrul acestui capitol se prezintă cele mai importante contribuții aduse, principalele dezvoltări pentru viitor la nivelul aplicației soft cât și la nivel hardware, precum si concluziile finale. Sintetizarea dezvoltărilor pentru viitor sunt necesare în vederea tranziției acestui model de la faza de laborator la faza de implementare pentru mașinile unei întreprinderi productive.

8.1 Concluzii finale

În lucrarea de față se abordează domeniul de mare actualitate al accesului controlat la resurse, cu particularizare la problematica autorizării.

Lucrarea este concepută ca o dezvoltare progresivă a problematicii accesului controlat la resurse bazat pe calificative, care se completează continuu, sfârșitul fiecărei etape evidențiind particularități și concluzii utilizabile în etapele ulterioare.

Obiectivul principal al lucrării îl constituie conceperea unei noi metode de acces controlat la resurse bazată pe calificative, prin intermediul căreia să se poată acorda angajaților unei compani dreptul de a accesa intrările anumitor obiecte protejate (mașini) în funcție de calificativele (aptitudinile) de care aceștia dispun.

Metoda de acces s-a formalizat printr-un pattern denumit QBAC (menționat și de asociația pattern-urilor pentru securitate [135]). În acest mod soluția propusă prin intermediul diagramelor de clase UML și însoțită de un exemplu din lumea reală, poate fi folosită și în alte situații similare. Totodată, pattern-ul QBAC poate fi combinat cu alte pattern-uri în vederea obținerii unor soluții mai complexe. Un exemplu de combinare al patternu-lui QBAC cu pattern-ul RBAC necesară pentru oferirea soluțiilor moderne de învățare, a fost prezentată în capitolul 3.

Pattern-ul QBAC îmbină două problematici actuale: accesul controlat la resurse fizice și necesitatea realizării unui sistem ce determină și totodată oferă angajaților posibilitatea de participare la un proces continuu de învățare.

Elaborarea fazei de dezvoltare a pattern-ului QBAC folosind Data Mining și diagrame UML, are ca scop descrierea algoritmilor și a funcționalităților prin intermediul unui limbaj formal, astfel încât metoda de acces propusă să poată fi implementată folosind orice limbaj de programare. Pentru implementarea acestei metode se recomandă o platformă ERP, datorită faptului că necesită o largă gamă de unelte, precum și de:

- crearea datelor angajaților;
- crearea structurii organizatorice a unei companii;

- calificative și cursuri.

Pentru comunicarea datelor între server (Platforma SAP NetWeaver) și PLC s-a realizat o codare binar – integer. În acest mod se evită vehicularea adreselor fizice ale intrărilor pentru care un subiect are drept de acces. Acestă codare este avantajoasă deoarece nu se încarcă rețeaua de comunicare, ajutând totodată și în procesul de integrare a noi obiecte protejate și intrări ale obiectelor protejate. Prin realizarea codării nu trebuiesc făcute modificări în variabilele globale ale PLC-ului, de fiecare dată când se adaugă noi obiecte protejate sau noi intrări unui obiect protejat. Dacă aceste modificări ar fi trebuit să fie realizate de fiecare dată când apare o modificare, s-ar fi consumat foarte mult efort din partea unui programator, acesta trebuind în acest caz să modifice variabilele globale PLC și aliace-urile din OPC server. Așadar, prin codarea realizată se minimizează pe cât posibil efortul necesar implementării schimbărilor aduse la nivel de obiect protejat și nu se încarcă rețeaua de comunicare dintre PLC și server.

Pentru realizarea comunicării dintre platforma SAP Netweaver și PLC s-au propus două soluții. Există firme specializate care oferă aplicații soft de legătură dintre platforma SAP și un PLC, dar aceste aplicații sunt scumpe și nu se oferă versiuni trial în vederea testării. De accea s-au dezvoltat propriile soluții folosind în acest scop tehnologia standard OPC (versiune server și client).

Utilizarea platformei de integrare și aplicații SAP NetWeaver pentru implementarea algoritmilor și funcționalităților metodei de acces create, ajută în accelerarea procesului de implementare, oferind totodată toate uneltele de care este nevoie. Totodată, prin intermediul uneltelor standard puse la dispoziție de această platformă, se oferă posibilitatea de a internaționaliza aplicațiile și obiectele de dezvoltare create. Aplicația realizată oferă suport pentru limba engleză și germană, dar extinderea suportului și pentru alte limbi nu necesită recodare, ci doar traducerea unor stringuri centralizate.

Prin utilizarea acestei platforme se oferă posibilitatea de extindere al bazelor de date generate de către modulul SAP ERP HCM cu bazele de date proprii, create în ABAP Dictionary. În acest mod, se pot folosi, acolo unde este posibil uneltele standard SAP pentru generarea bazelor de date necesare, reducându-se astfel numărul de tabele ce trebuiesc create manual în ABAP Dictionary.

Totodată, prin folosirea acestei platforme nu este necesară crearea propriei aplicații de administrare al datelor angajaților, al calificativelor, al cursurilor putându-se folosi screen-urile SAP standard. Aplicația de administrare ce trebuie realizată indiferent de mediul în care se implementează metoda de acces, este aplicația ce administrează obiectele protejate și intrările acestora.

Metoda de acces astfel implementată, poate fi oferită ca și o extra funționalitate SAP, ce extinde modulul standard AP ERP HCM.

Pentru implementarea patern-ului QBAC s-a folosit un sistem distribuit ce a fost studiat din punct de vedere al îndeplinirii cerințelor unui astfel de sistem.

Concluzionând, putem spune că s-a obținut un sistem transparent din punct de vedere al transparenței erorilor, un sistem deschis, un sistem ce necesită dezvoltări ulterioare pentru asigurarea securității. În ceea ce privește scalabilitatea și concurența acestea se vor studia în versiuni viitoare, versiuni în care se va dispune de mai multe mașini de test.

În ceea ce privește securitatea sistemului, trebuie să se țină cont de faptul că folosirea unei metode de autentificare cu grad ridicat de siguranță și a autorizării entității autentificate nu înseamnă că s-a obținut un sistem sigur, acestea trebuind combinate cu o programare sigură care să înlăture eventualele atacuri precum și cu alte metode, ca de exemplu criptografia.

Necesitatea dezvoltării acestei metode de acces nu este una pur teoretică ci este o necesitate ce vine din nevoia concretă a sistemelor productive. După dezvoltări viitoare și retestări în vederea îndeplinirii cerințelor principale ale unui sistem distribuit, această metodă se va implementa în producție, câteva firme fiind interesate de o viitoare implementare.

Colaborarea cu firme specializate în dezvoltarea modulelor de autentificare va duce la creșterea securității sistemului distribuit necesar implementării QBAC, precum și la posibilități de adaptare în funcție de necesitățile fiecărui client.

8.2 Contribuții aduse

Pornind de la obiectivele declarate ale lucrării de față, în continuare sunt prezentate principalele contribuții originale:

- Elaborarea unei sinteze critice asupra stadiului actual al accesului controlat la resurse.
- Analizarea importantei metodelor de acces la resurse.
- Conceperea și dezvoltarea unei metode de acces controlat la resurse fizice bazată pe calificative, metodă ce are ca și principale avantaje:
    - obținerea unui model prin care se integrează obiectele protejate în sfera resurselor umane, putându-se raspunde la întrebări de genul: "Ce calificative sunt necesare unui angajat pentru a putea îndeplini job-ul de instalator la mașina x.y?";
    - obținerea unui model dinamic, flexibil și maleabil la schimbări
    - susține procesul de învățare continuu al angajaților;
    - posibilitate de combinare a pattern-ului QBAC cu pattern-ul RBAC pentru oferirea soluțiilor moderne de învățare.
- Conceperea și dezvoltarea unui pattern pentru accesul controlat la resurse fizice numit QBAC (Qualifications Based Access Control).
- Conceperea și dezvoltarea unui politici de autorizare.
- Elaborarea fazei de dezvoltare al pattern-ului QBAC folosind Data Mining și diagrame UML.
- Analizarea cerințelor sistemului informatic necesar implementării QBAC.
- Prezentarea modului în care pattern-ul QBAC poate fi extins cu pattern-ul RBAC.
- Specificarea structurii sistemului distribuit necesar implementării pattern-ului QBAC.
- Determinarea unui model matematic al sistemului distribuit.
- Elaborarea unei logici de codare intre client si server.
- Implementarea și validarea experimentală, a algoritmilor propuși pentru accesul controlat la resurse bazat pe calificative.
- Analizarea avantajelor prin care platforma de integrare și aplicații SAP NetWeaver ajută la implementarea pattern-ului QBAC.

- Îmbinarea bazelor de date proprii cu bazele de date generate de componenta standard SAP ERP HCM în vederea obținerii unor rezultate maximale pentru implementarea QBAC.
- Internaționalizarea aplicațiilor realizate, oferind suport pentru limba germană și engleză.
- Analizare posibilităților de comunicare dintre platforma SAP NetWeaver și tehnologia RFID.
- Propunerea a doua soluții în vederea comunicării dintre Application Server ABAP și un PLC.
- Realizarea unui stand de test echipat cu un PLC SIMATIC S7-300, o mașină de test, un citor RFID și un panou de comandă în vederea testării metodei de acces controlat propusă.
- Elaborarea programelor destinate implementarii si analizei pattern-ului QBAC.
- Extinderea modulului standard SAP ERP HCM.

8.3 Dezvoltări pentru viitor

Soluția propusă în prezenta teză este o soluție viabilă în domeniul accesului controlat la resurse, necesitând dezvoltări ulterioare în vederea tranziției de la varianta de test QBAC la o variantă ce va putea fi implementată, ca și primă versiune într-un sistem productiv.

Principale dezvoltări ce trebuiesc realizate pentru viitor sunt:

- Dezvoltări la nivel **hardware**:
  - Extinderea numărului de mașini de test folosite și retestare la nivel de model.
  - Studierea mașinilor dintr-o întreprindere reală în vederea realizării unei structuri concrete.
  - Studierea eventualelor excepții și situații speciale ce trebuiesc luate în considerare la implementarea pe mașini reale.
  - Studierea necesităților hardware aferente.
  - Crearea unei rețele de PLC necesare implementării metodei de acces pentru n mașini ale unei întreprinderi.
  - Studierea modului în care QBAC pentru accesul la intrările unor obiecte protejate poate fi extins cu operații de genul acces la zone securizate, pontaj automat sau acces la parcarea firmei.
  - Colaborări cu firme specializate în autentificare (ex. Firma PCS) astfel încât să se poată oferi clienților pentru QBAC module de autentificare ce pot fi customizate în funcție de necesitățile acestora. În acest sens diferitele module de autentificare oferite de către o astfel de firmă vor putea fi îmbinate în logica QBAC acoperind o largă gamă de tehnici de autentificare de la cea biometrică până la simple carduri de proximitate.

- Dezvoltări la nivelul aplicației **soft**:
  - În cazul mașinilor unei întreprinderi studierea și implementarea eventualelor priorități la nivel PLC pentru intrările aferente mașinilor.

- După implementarea noului model restudierea vulnerabilităților și a rezistenței la atacuri.
- Eventuale înlocuiri ale SQL cu folosirea claselor persistente.
- Extinderi la nivelul portalului firmei.
- Extinderi software în vederea introducerii diverselor module de autentificare în logica QBAC (în cazul colaborării cu o firmă specializată în autentificare).
- Adăugarea semnăturilor digitale la Web Service-ul de comunicare între platforma SAP NetWeaver și PLC (în cazul în care se folosește această variantă).
- Pentru cazul în care se păstrează cardurile RFID: Realizarea programului care să tipărească cardurile RFID dispunând de anumite mostre de formatare și folosind datele direct din modulul de resurse umane.
- Analizarea scalabilității și concurenței sistemului distribuit rezultat după extinderea numărului de mașini.

Anexe

A1. Rezultate obținute pe parcursul stagiului doctoral

- **Cărți publicate sau în curs de publicare pe parcursul stagiului doctoral**

- Ulrich Gellert și Ana Daniela Cristea, ***Praxishandbuch Web Dynpro ABAP***, Editura Springer Berlin, Septembrie 2010, ISBN: 978-3-642-11386-4

- Ulrich Gellert și Ana Daniela Cristea, ***Web Dynpro ABAP for Practitioners***, Editura Springer Berlin, Iulie 2010, ISBN: 978-3-642-11384-0

- Cristea Ana Daniela și Pănoiu Caius, **Interfețe și periferice**, Editura Mirton Decembrie 2006, ISBN: 978-973-52-0004-6

- **Lucrări publicate sau acceptate la jurnale, conferințe și simpozioane ISI, IEEE (ca și prim autor sau coautor)**

- Cristea Ana Daniela, Adela Diana Berdie, Osaci Mihaela și Chirtoc Vasile Daniel, ***The Advantages of using Mind Map for learning Web Dynpro***, Jurnal: Computer Applications in Engineering Education, acceptată 26 Septembrie 2008, publicată online în 19 Feb 2009 (Jurnal ISI, factor de impact 0,3l CSA/CIG), disponibil online la: http://www3.interscience.wiley.com/journal/122210203/abstract?CRETRY=1&SRETRY=0

- Cristea Ana Daniela, Octavian Prostean, Thomas Muschalik și Tirian Ovidiu, ***Distributed system for access control to physical resources based on qualifications***, WSEAS 2010, SEPADS'10 (SOFTWARE ENGINEERING, PARALLEL and DISTRIBUTED SYSTEMS) International Conference, University of Cambridge, Cambridge, UK, Februarie 20-22 2010, ISBN: 978-960-474-156-4, ISSN: 1790-5117, (conferință ISI proceeding).

- Cristea Ana Daniela, Octavian Prostean, Mushalik Thomas și Ovidiu Tirian, ***Using SAP NetWeaver to implement a new auhorization concept based on qualifications and physical connection through RFID***, The 3rd European DAAAM International Young Researchers´ and Scientists´ Conference 25-28th November 2009, Vienna, Austria, ISBN 978-3-901509-70-4, ISSN 1726-9679, (conferință ISI proceeding).

- Cristea Ana Daniela, Octavian Prostean, Mushalik Thomas și Ovidiu Tirian, ***An Access Control Pattern based on Qualifications to Grand Access***

to Physic Resources, The 3rd European DAAAM International Young Researchers´ and Scientists´ Conference 25-28th November 2009, Vienna, Austria, ISBN 978-3-901509-70-4, ISSN 1726-9679, (conferință ISI proceeding).

- Cristea Ana Daniela și Octavian Prostean, ***Dynamic programming with Web Dynpro ABAP***, 5th International Symposium on Applied Computational Intelligence and Informatics, SACI-2009, May 28–29, 2009 – Timişoara, Romania (Coferință indexată ISI).

- Cristea Ana Daniela, Octavian Prostean, Thomas Muschalik și Tirian Ovidiu, ***Contribution to the creation and development of a new authorization concept based on a learning process***, Jurnal: Computer Applications in Engineering Education, acceptată 07 Iulie 2010, (Jurnal ISI, factor de impact 0,3l CSA/CIG).

- Tirian Ovidiu, Octavian Prostean, Rusu-Anghel Stela, Pinca-Bretotean Camelia și Cristea Ana Daniela, ***Fuzzy system for implementing the cracks control during the continuous casting***, The 3rd European DAAAM International Young Researchers´ and Scientists´ Conference 25-28th November 2009, Vienna, Austria, ISBN 978-3-901509-70-4, ISSN 1726-9679, (Conferință ISI Proceeding).

- Tirian Ovidiu, Camelia Bretotean Pinca, Daniela Cristea și Marcel Topor, ***Research on the elimination of cracks in continuous casting plan using fuzzy logic***, WSEAS, Recent Advances in Circuits, Systems, Electronics, Control and Signal Processing, ISBN 978-960-474-139-7, ISSN 1790-5117, pagini 273-278 (Conferință ISI Proceeding).

- Cristea Ana Daniela, Octavian Prostean, Muschalik Thomas și Tirian Ovidiu, **Development objects and algorithms required to implement a new method of access control to physical resources, based on qualifications**, Articol nr 89-478, acceptată spre publicare în jurnal ISI WSEAS ca urmare a unei invitații speciale primate de la conferința SEPADS'10, dar aflată încă în fază de review (Jurnal ISI).

- Cristea Ana Daniela, Octavian Prostean, Muschalik Thomas și Tirian Ovidiu, ***The advantages of using SAP NetWeaver platform to implement a multidisciplinary project***, acceptată la ICCC-CONTI 2010, 27-29 Mai, Timișoara România (Conferință IEEE).

- Tirian Ovidiu, Prostean Gabriela, Stela Rusu – Anghel, Cristea Ana Daniela, **Adaptive control system of continuous casting process based on a fuzzy logic mechanism**, ICCC-CONTI 2010, 27-29 Mai, Timișoara România (Conferință IEEE).

- Cristea Ana Daniela, Berdie Adela și Mihaela Osaci, **Parallel between two SAP Frameworksthat use MVC paradigm**, ICCC-CONTI 2010, 27-29 Mai, Timișoara România (Conferință IEEE).

➢ **Lucrări publicate în jurnale, conferințe naționale și internaționale SDN, precum și indexate BDI**

- Cristea Ana Daniela, ***Messages internationalization with Web Dynpro ABAP***, publicat online pe seit-ul SDN, acceptat 01 Mai 2009. Disponibil online la: (comunitatea SAP Help)
  https://www.sdn.sap.com

- Cristea Ana Daniela, ***Multilanguage Qualification Catalogue used in a WebDynpro Application***, publicat online pe seit-ul SDN (Comunitatea SAP Help), acceptat 01 Septembrie 2009. Disponibil online la: (comunitatea SAP Help)
  http://www.sdn.sap.com

- Cristea Ana Daniela, Adela Diana Berdie și Osaci Mihaela, ***Working with ABAP Persistent data***, International Symposium "INTERDISCIPLINY REGIONAL RESEARCH", Hunedoara 23 - 24 Aprilie 2009, ISIRR 2009.

- Cristea Ana Daniela și Adela Diana Berdie, ***Error handling and messages with application Server ABAP***, International Symposium "INTERDISCIPLINY REGIONAL RESEARCH", Hunedoara 23 - 24 Aprilie 2009, ISIRR 2009

- Cristea Ana Daniela, Adela Berdie și Mihaela Osaci, ***User interfaces with Web Dynpro ABAP and Web Dynpro Java***, The Knowledge-based organization, The 14th International Conference, Technical Sciences Computer Science, Modeling & Simulation and E-Learning Technologies, Sibiu 27-29 NOVEMBER 2008, Pag. 157 – 164.

- Cristea Ana Daniela, Berdie Adela și Osaci Mihaela, ***The use and importance of Hook Methods in Web Dynpro ABAP and Web Dynpro Java***, SCIENTIFIC BULLETIN of "Politehnica" University of Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol. 54 (68), Fasc. 3, 2009, ISSN 1224-600X, (Jurnal categoria B+, indexat BDI).

- Cristea Ana Daniela, ***Web Dynpro ABAP Presentation through a Mind Map***, Jurnal of Engineering annals of Faculty engineering Hunedoara TOME VI, 2008, Fascicula 3, ISSN 1584 – 2673, Pag 194 – 199 (Jurnal categoria B+, indexat BDI).

- Cristea Ana Daniela, ***ABAP Dictionary used in Web Dynpro application***, Jurnal of Engineering annals of Faculty engineering Hunedoara TOME VI 2008, Fascicula 3, ISSN 1584 – 2673, Pag 222 – 227 (Jurnal categoria B+, Indexat BDI).

- Cristea Ana Daniela, Berdie Adela și Osaci Mihaela, ***Application componentization and component usages with Web Dynpro ABAP and Web Dynpro Java***, SCIENTIFIC BULLETIN of "Politehnica" University of

Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol. 55 (69), Fasc. 1, Martie 2010, ISSN 1224-600X, (Jurnal categoria B+, indexat BDI).

➢ **Alte rezultate și activități**

- Lucrarea prezentată în cadrul conferinței desfășurate la University of **Cambridge UK** a fost selectată între cele **30%** cele mai bune lucrări prezente la conferință.

- Membru în organizatia **Security Patterns**, pattern-ul **QBAC** dezvoltat fiind **recunoscut și menționat** ca și contribuții aduse în acest domeniu [137].

- Lucrarea *"Messages internationalization with Web Dynpro ABAP"* a fost publicată pe prima pagină a topicului Web Dynpro ABAP pe seit-ul **SAP help**.

- Firma NWCON în cadrul căreia s-a realizat acest proiect, a participat în anul 2009 cu acest proiect la concursul **Hessen Champions** ocupând locul 8 între cele 1100 cele mai inovative firme din Hessen (http://www.hessen-champions.de).

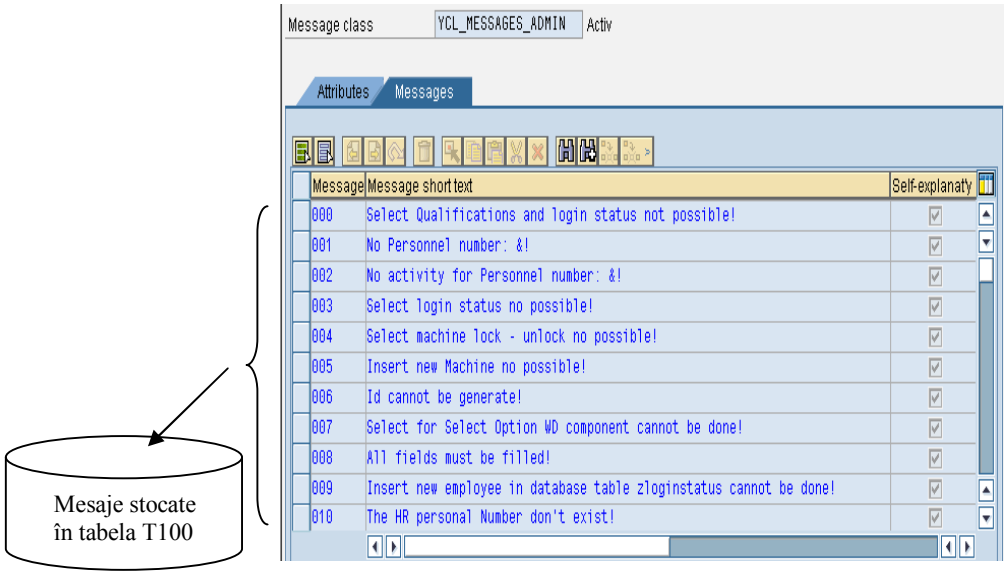
A2. Exemplu de codare a unei metode din clasa YCL_LOGIN_SESSION

```
method qualification.
  data itab type standard table of zqualifications.
  data:  begin of st_qual,
         objid type objid,
         end of st_qual.
  data qual_tab like standard table of st_qual.
  data lv_endda type hrp1000-endda.
  field-symbols <fs_endda> type any.
  assign lv_endda to <fs_endda>.
  data : it_qual like standard table of st_qual.
  data t_result like table of it_qual.
  field-symbols: <itab_fs> like line of itab,
                 <tqual_fs> like line of it_qual.
  try.
    select  *
    from zqualifications
    into table itab where machine_id = machine_id.
    if sy-subrc <> 0.
      raise exception type zcx_excep_nasapcfrfid
      exporting
        textid = zcx_excep_nasapcfrfid=>zcx_select_no_posible.
    endif.
  endtry.
```

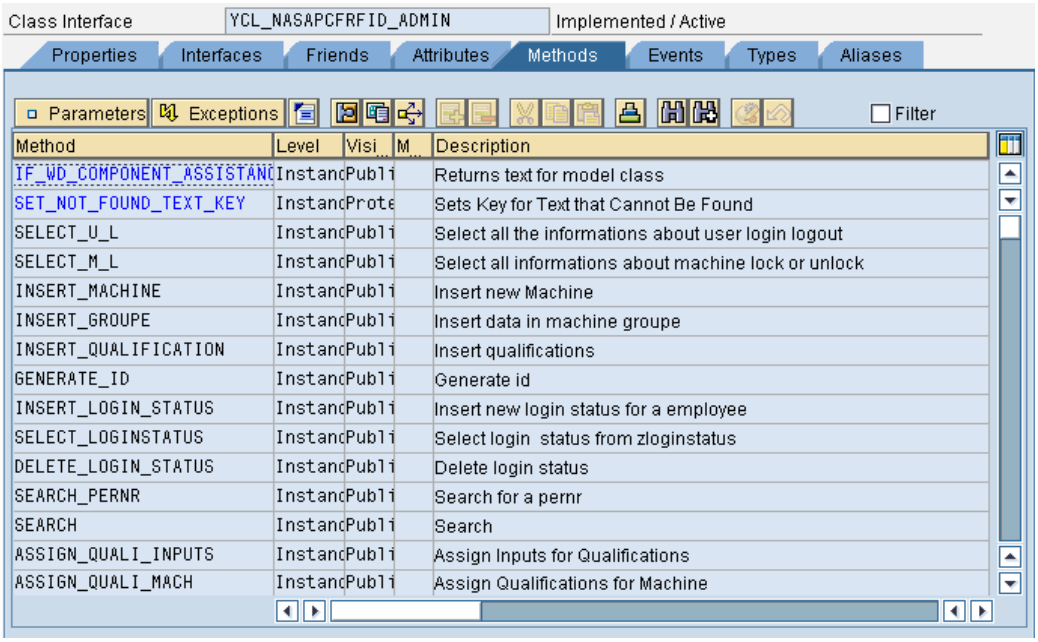
```
  data lv_stext type hrp1000-stext.
  field-symbols <fs_stext> type any.
  assign lv_stext to <fs_stext>.
  loop at itab assigning <itab_fs>.
    select  b~objid into table it_qual
                        from hrp1000 as a
                        inner join hrp1001 as b on a~objid = b~objid and b~sobid = per
nr
                        where a~otype = 'Q'
                          and a~plvar = '01'
                          and a~istat = 1
                          and a~langu = sy-langu
                          and a~objid = <itab_fs>-objid
                          and b~otype = 'Q'
                          and b~plvar = '01'
                          and b~istat = 1
                          and b~sclas = 'P'.
    if sy-subrc eq 0.
      loop at it_qual assigning <tqual_fs> .
        select single endda from hrp1000 into <fs_endda> where
          objid = <tqual_fs>.
        if <fs_endda> ge sy-datum.
          insert <tqual_fs> into table qual_tab.
        else.
          select single stext from hrp1000 into <fs_stext> where
 objid = <tqual_fs> and endda = <fs_endda> and langu = sy-langu.
          concatenate expire_message <fs_stext> into expire_message.
        endif.
      endloop.
    endif.
  endloop.
  data:  begin of st_inputs,
         inputs type zbuttons-button_code,
         end of st_inputs.
  field-symbols: <fs_qual_tab> like line of qual_tab.
  data input_tab like standard table of st_inputs.
  loop at qual_tab assigning <fs_qual_tab>.
    select button_code from zvquali_button into table input_tab where objid = <fs_q
ual_tab>
      and machine_id = machine_id.
    field-symbols: <fs_input_tab> like line of input_tab.
    data:  begin of st_inputs2,
           inputs type zbuttons-button_code,
           end of st_inputs2.
    data input_code_tab like standard table of st_inputs2.
    loop at input_tab assigning <fs_input_tab>.
      insert <fs_input_tab> into table input_code_tab.
      sort input_code_tab by inputs.
      delete adjacent duplicates from input_code_tab comparing inputs.
    endloop.
```

```
  endloop.
  field-symbols: <fs_input_code_tab> like line of input_code_tab.
  data exp type f value 0.
  data s type i value 0.
  data nr_elements type i.
  data lv_arg2 type f.
  loop at input_code_tab assigning <fs_input_code_tab>.
    lv_arg2 = <fs_input_code_tab>-inputs.
    lv_arg2 = lv_arg2 - 1.
    cl_foev_builtins=>power( exporting im_arg1 = 2
                                       im_arg2 = lv_arg2
                             importing ex_result = exp ).
    s = s + exp.
  endloop.
  qualification = s.
endmethod.
```

A3. Exemplu clasă de mesaje creată, textele acesteia fiind folosite în clasa de excepții YCX_EXCEPTION_ADMIN



A4. Clasă de asistență (26 metode) folosită ca și model pentru aplicația de administrare



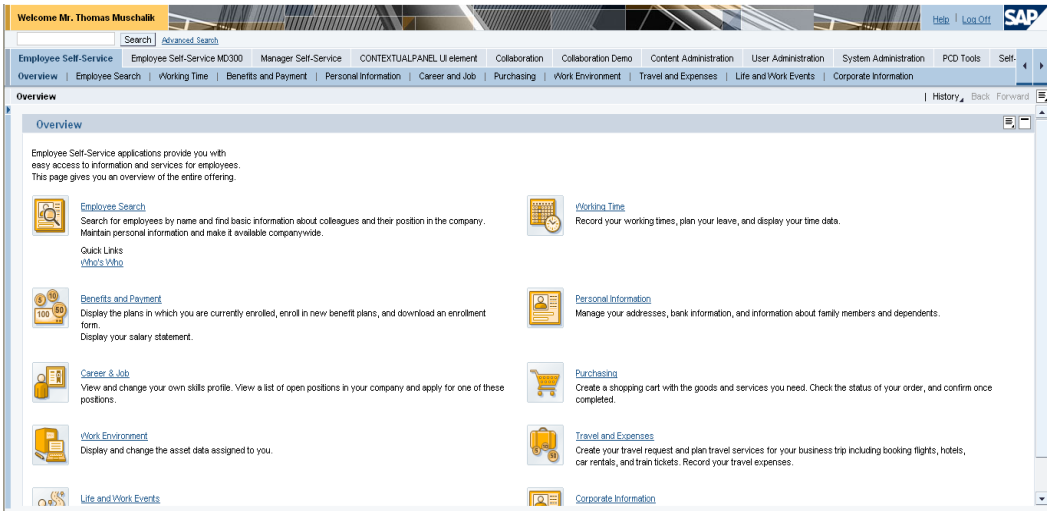
A5. Exemplu de codare a unei metode de tratare a evenimentelor din aplicația creată în Web Dynpro ABAP

```
method onactionnext .
  data lo_nd_loginstatus type ref to if_wd_context_node.
  data lo_el_loginstatus type ref to if_wd_context_element.
  data ls_loginstatus type wd_this->element_loginstatus.
  data lv_pernr type zloginstatus-pernr.
  lo_nd_loginstatus = wd_context->get_child_node( name = wd_this-
>wdctx_loginstatus ).
  lo_el_loginstatus = lo_nd_loginstatus->get_element(  ).
  lo_el_loginstatus->get_attribute(
    exporting
      name =  `PERNR`
    importing
      value = lv_pernr ).
  data lr_node type ref to if_wd_context_node.
  data lr_oref type ref to ycx_exceptions_admin.
  data ls_data type if_v_main=>element_btn_enabled.
  data lv_selectedstep type string.
  lr_node = wd_context->get_child_node( 'BTN_ENABLED' ).
  lr_node->get_attribute( exporting name = 'SELECTEDSTEP'
                          importing value = lv_selectedstep  ).
  case lv_selectedstep.
    when 'SELECT'.
      if lv_pernr is not initial.
```

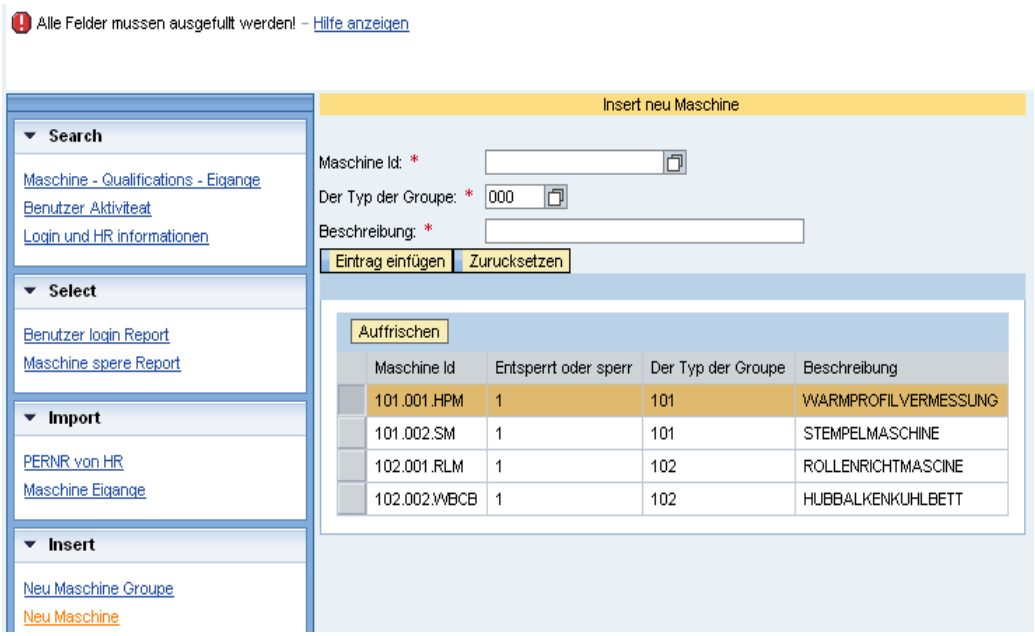
```
    data bool type i.
    wd_assist->search( exporting pernr = lv_pernr
    importing  bool = bool ).
    if bool = 2.
      ls_data-selectedstep = 'DELETE'.
      ls_data-btn_next = abap_true.
      ls_data-btn_prev = abap_true.
      data: l_node type ref to if_wd_context_node.
      l_node = wd_context->get_child_node( name = `HRPLOGIN` ).
      data lt_hrlogin type standard table of z_hrlogin.
      try.
          wd_assist->search_pernr( exporting pernr = lv_pernr
               importing return = lt_hrlogin ).
          l_node->bind_elements( lt_hrlogin ).
        catch ycx_exceptions_admin into lr_oref.
          wd_this->create_usage( ).
          wd_this->raise_exception( exporting p_lr_oref = lr_oref ).
      endtry.
      wd_this->fire_op_to_v_step2_plg(
      ).
    else.
      try.
          raise exception type ycx_exceptions_admin
          exporting textid = ycx_exceptions_admin=>ycx_pernr_no_exist
                          pernr = lv_pernr.
        catch ycx_exceptions_admin into lr_oref.
          wd_this->create_usage( ).
          wd_this->raise_exception( exporting p_lr_oref = lr_oref ).
          exit.
      endtry.
    endif.
  else.
    try.
        raise exception type ycx_exceptions_admin
        exporting textid = ycx_exceptions_admin=>ycx_pernr_mandatory.
      catch ycx_exceptions_admin into lr_oref.
        wd_this->create_usage( ).
        wd_this->raise_exception( exporting p_lr_oref = lr_oref ).
        exit.
    endtry.
  endif.
when  'DELETE'.
  ls_data-selectedstep = 'SHOW'.
  ls_data-btn_next = abap_false.
  ls_data-btn_prev = abap_true.
  try.
      wd_assist->delete_login_status( exporting pernr = lv_pernr ).
    catch ycx_exceptions_admin into lr_oref.

      wd_this->create_usage( ).
      wd_this->raise_exception( exporting p_lr_oref = lr_oref ).
```

```
        exit.
      endtry.
      wd_this->fire_op_to_v_step3_plg( ).
    when others.
      ls_data-btn_next = abap_false.
      ls_data-btn_prev = abap_true.
  endcase.
  lr_node->set_static_attributes( ls_data ).
endmethod.
```

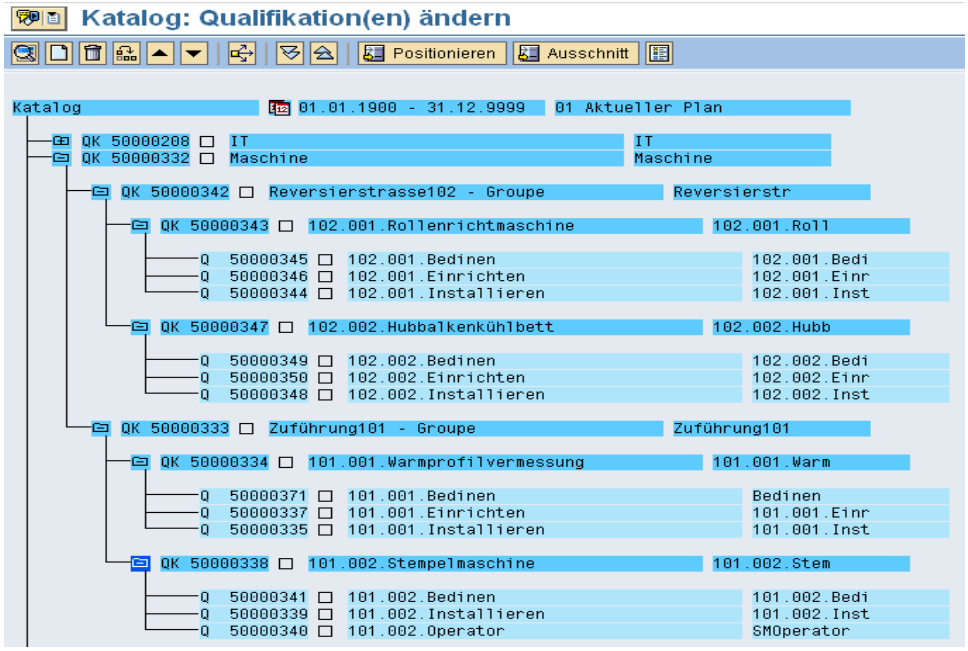
A6. Portal – captura cu ESS



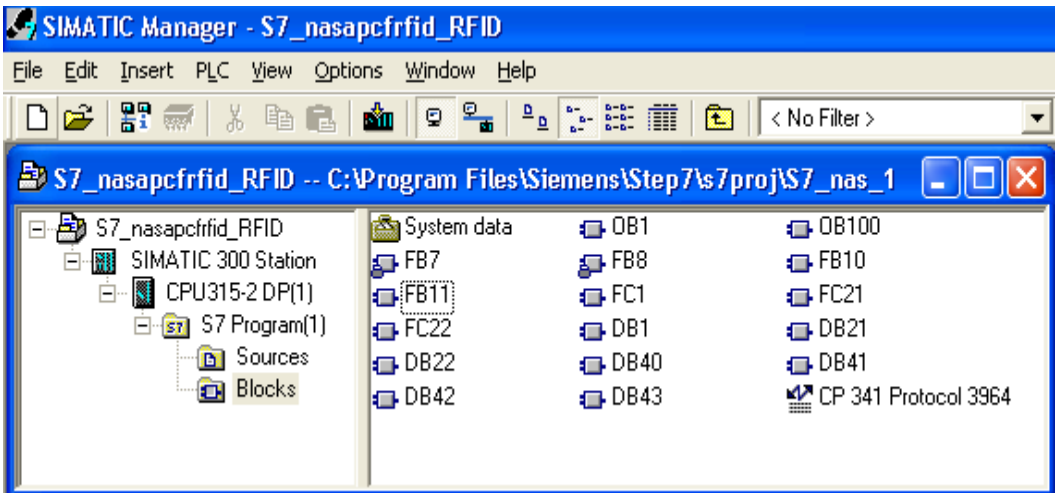
A7. Captură din aplicația de administrare – limbă de logare germană



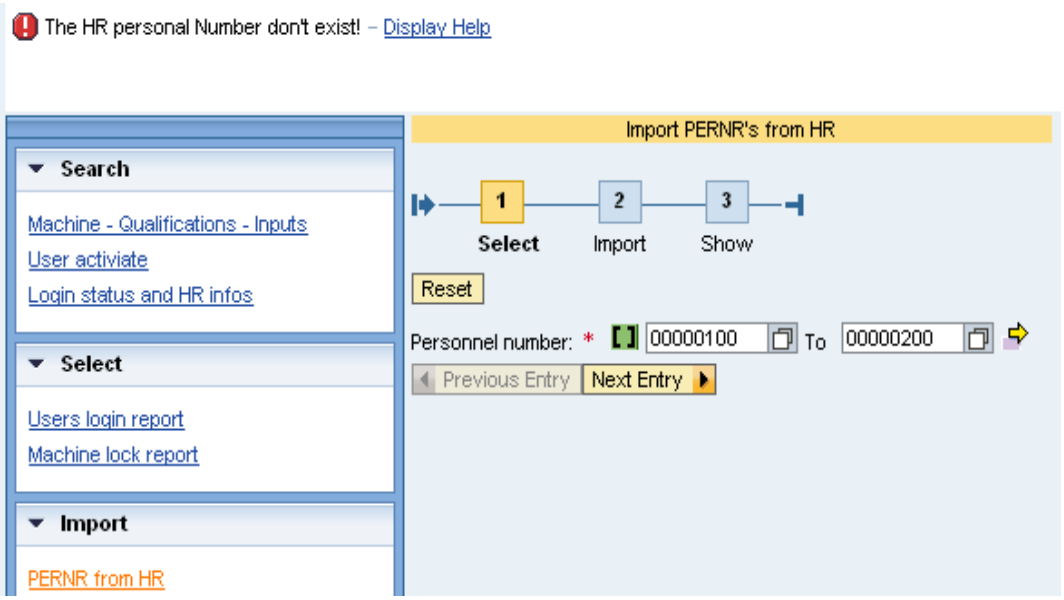
A8. Catalog de calificative – limbă germană

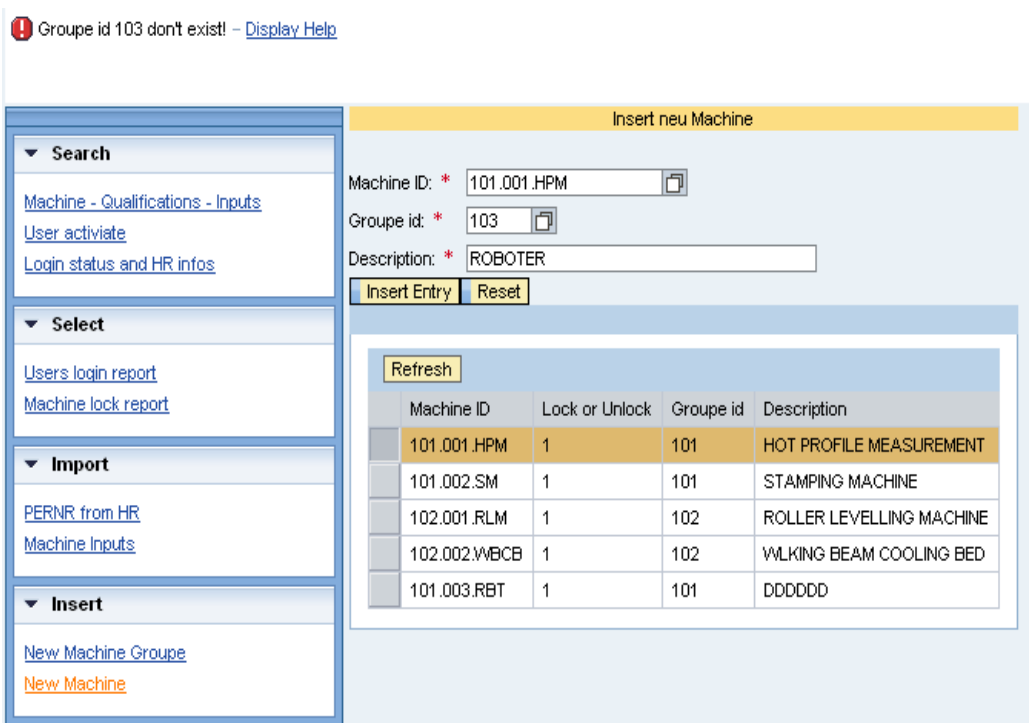


A9. Structura proiectului Step7 rezultat



A10. Transparența erorilor la nivelul aplicației de administrare




Groupe id 103 don't exist! – Display Help
Search
Machine - Qualifications - Inputs
User activiate
Login status and HR infos
Select
Users login report
Machine lock report
Import
PERNR from HR
Machine Inputs
Insert
New Machine Groupe
New Machine
Insert neu Machine
Machine ID: * 101.001.HPM
Groupe id: * 103
Description: * ROBOTER
Insert Entry
Reset
Refresh
Machine ID
Lock or Unlock
Groupe id
Description
101.001.HPM 1 101 HOT PROFILE MEASUREMENT
101.002.SM 1 101 STAMPING MACHINE
102.001.RLM 1 102 ROLLER LEVELLING MACHINE
102.002.WBCB 1 102 WLKING BEAM COOLING BED
101.003.RBT 1 101 DDDDDD

Bibilografie

[1] – E. Byres și D. Hoffman, The myths and facts behind cyber security risks for Industrial Control Systems, VDE Congres 2004, disponibil online: http://www.isa.org/CustomSource/ISA/Div_PDFs/PDF_News/Glss_2.pdf

[2] - Introduction to Business Security Patterns An IBM White Paper, Business Security Patterns Overview, disponibil online: http://www-03.ibm.com/security/patterns/intro.pdf

[3] – Bogdan Ioan Groza, Contributii criptografice hibride, bazate pe tehnici simetrice si asimetrice – aplicatii in sisteme de conducere, teza de doctorat, Seria 1 : Automatica, nr.11, Editura Politehnica, ISSN : 1842-5208, ISBN : 978-973-625-688-2

[4] - Mark Stamp, Information security, Principles and Practice, Editura JOHN WILEY & SONS, INC., 2006, ISBN 978-0-471-73848-0

[5] – Schumacher M., Rodig U și Moschgath L, Hacker Contest, Springer 2003, ISBN: 3-540-41164-X

[6] - Rabia Sirhindi, Asma Basharat și Ahmad Raza Cheema, Depth-in-Defense Approach against DDoS, 6th WSEAS International Conference on Information Security and Privacy, Tenerife, Spain, December 14-16, 2007

[7] - Stuart McClure, Joel Scambray, George Kurtz, Securitatea retelelor, Editura Teora, ISBN : 973-20-0300-6

[8] - Știri și articole realizate de către Antena3, disponibile online la: http://www.antena3.ro/etichete/atac_informatic/

[9] - E. B. Fernandez, "An Overview of Internet Security", presented at World's Internet and Electronic Cities Conference (WIECC 2001), Kish Island, Iran, 2001.

[10] - Markus Schumacher, Security Engineering with patterns, Origins, Theoretical Model and New Applications, 978-3540407317, Springer 2003

[11] - Gao, Critical infrastructure Protection, Challenges and efforts to secure control systems, United States General Accountig Office, Martie 2004, disponibil online: http://www.gao.gov/new.items/d04354.pdf

[12] - ABBASS ASOSHEH şI NAGHMEH RAMEZANI, A New and Comprehensive Taxonomy of DDoS Attacks and Defense Mechanism, 6th WSEAS International Conference on Information Security and Privacy, Tenerife, Spain, December 14-16, 2007

[13] – ALI GHAFFARI, Vulnerability and Security of Mobile Ad hoc Networks, Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, Lisbona, Portugalia, Septembrie 22-24, 2006

[14] – Statistici CERT, disponibile online: http://www.cert.org/stats/cert_stats.html#vulpubs

[15] - 2008 CSI Computer Crime and Security Survey, disponibil online: http://www.cse.msstate.edu/~cse6243/readings/CSIsurvey2008.pdf

[16] – Kabay M, Understanding Studies and Surveys of Computer Crime, Iunie 2009, disponibil online: http://www.mekabay.com/methodology/crime_stats_methods.pdf

[17] - Robert Willison, Motivations for Employee Computer Crime: Understanding and Addressing Workplace Disgruntlement through the Application of Organisational Justice
disponibil online la:

http://openarchive.cbs.dk/bitstream/handle/10398/7759/WP_2009_001.pdf?sequence=3

[18] – Christopher Alexander, Sara Ishikawa, Murray Silversein, A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series), Oxford University Press 1977

[19] – Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design pattern elements of reusable Object Oriented software, ISBN: 978-0201633610, Addison – Wesley, 1995

[20] - Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, Peter Sommerland, Security Patterns Integrating Security and Systems Engineering, John Wiley & Sons, Ltd, 2006

[21] - Pierangela Samarati şi Sabrina De Capitani di Vimercati, Access Control: Policies, Models, and Mechanisms, Capitol din cartea: Foundations of Security Analysis and Design), ISBN: 978-3-540-42896-1, Editura Springer Berlin / Heidelberg, paginile 38-47, Volume 2171/2001

[22] - Alan Shalloway şi James R. Trott, Design Patterns Explained, a new perspective on Object Oriented Design, Software patterns series, ISBN: 978-0201715941, Editura Addison-Wesley Professional, prima ediţie (Iulie 9, 2001)

[23] - Sherif M. Yacoub și Hany H. Ammar, Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems, ISBN: 0-201-77640-5, Editura Addison-Wesley 2003

[24] - Eric Gamma, Richard Helm, Ralph Johnson și John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, ISBN: 978-0201633610, Editura Addison-Wesley 1994

[25] – Eduardo B. Fernandez, Günther Pernul și Maria M. Larrondo-Petrie, Patterns and Pattern Diagrams for Access Control (Capitol din cartea: Trust, Privacy and Security in Digital Business, Proceeding of 5th International Conference TrustBus 2008, Turin Italy, septembrie 2008), ISBN: 978-3-540-85734-1 , Editura Springer Berlin / Heidelberg, paginile 38-47, Volume 5185/2008

[26] – Fernandez, Eduardo B. and Pernul, Günther (2006) Patterns for session-based access control. In: Proceedings of the 2006 conference on Pattern language. ACM International Conference Proceeding Series . ACM, NewYork

[27] - David G. Rosado, Eduardo Fernandez-Medina, Mario Piattini, Comparison of Security Patterns, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.2B, February 2006

[28] – R. Sandhu et al: Role Based Access Control models, IEEE Computer, Vol. 29, No. 2, February 1996, pp. 38 – 47

[29] - Eduardo B. Fernandez, Metadata and authorization patterns, disponibil la: http://www.cse.fau.edu/~ed/MetadataPatterns.pdf

[30] - David G Rosado, Carlos Guitierrez, Eduardo Fernandez-Medina, Mario Piattini, A study of Security Architectural Patterns, Proceeding of the First International Conference on Availability, Reliability and Security, IEEE 2006

[31] - Eduardo B. Fernandez, Jose Ballesteros, Ana C. Desouza-Doucet, Maria M. Larrondo-Petrie , Security Patterns for Physical Access Control Systems, Capitol din cartea: Data and Applications Security XXI), Springer Berlin/Heidelberg, ISBN: 978-3-540-73533-5, Volume 4602/2007, pagini 259–274

[32] - Cristea Ana Daniela, Octavian Prostean, Thomas Muschalik și 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 November 2009, Vienna, Austria, ISSN 1726-9679, ISBN 978-3-901509-70-4, pagini: 1765 – 1766

[33] - G. Meszaros, Jim Doble, A pattern Language for Pattern Writing, Pattern Languages of Programs (PLoP) 1996

[34] – Software Visual Paradigm pentru crearea diagramelor UML, http://www.visual-paradigm.com/product/vpuml/

[35] - Dorin Bocu, Razvan Bocu, Modelarea Obiect Orientata cu UML, editura albastra 2006, Cluj

[36] - Martin Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, Ediția a treia, ISBN: 0-321-19368-7, 2003, Editura Addison Wesley

[37] - Thomas Erler, Das Einsteigerseminsar UML2, 2004, Editura: Moderne industrie Buch AG&Co. KG, Landsberg, ISBN 3-8266-7363-8

[38] - Sinan Si Alhir, Learning UML, ISBN: 0-596-00344-7, Editura O'Reilly, 2003

[39] - Michael Jesse Chonoles and James, UML2 for Dummies, ISBN: 978-0764526145, Editura John Wiley & Sons, 2003

[40] – Jiawei Han, Jiawei Han și Jian Pei Data Mining: Concepts and Techniques, Second Edition, Editura Morgan Kaufmann 2005, ISBN 978-1558609013

[41] – Ian H. Witten și Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Editura Morgan Kaufmann 2005, ISBN: 978-0120884070

[42] – Krista Rizman Zalik, *Learning through data mining*, Computer Applications in Engineering Education, Volumul 13, Publicat: 14 Aprilie 2005.

[43] - Cristea Ana Daniela, Octavian Prostean, Muschalik Thomas și Tirian Ovidiu, Development objects and algorithms required to implement a new method of access control to physical resources, based on qualifications, Articol nr 89-478, acceptată spre publicare la jurnal ISI WSEAS ca urmare a unei invitații speciale primate de la conferința SEPADS'10 dar aflată încă în faza de review

[44] – Cristea Ana Daniela și Pănoiu Caius, Interfețe și periferice, Editura Mirton Timișoara 2006, ISBN 978-973-52-0004-6

[45] – Grigor Moldovan, Ioan Dziac, Sisteme distribuite – modele matematice, Editura universității Angora Oradea, 2006, ISBN 978-973-87960-9-6

[46] – Wan Fokkink, Modelling Distributed Systems, Editura Springer 2007, ISBN: 978-3540739371

[47] – George Coulouris, Jean Dollimore și Tim Kindberg, Distribuited systems concept and design, ISBN 978-0201619188, Addison-Wesley, 2004

[48] – Gerard Tel, Introduction to Distributed Algorithms, Editura Cambridge University 2000, ISBN: 978-0521794831

[49] – Ajay D. Kshemkalyani și Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Editura Cambridge University Press 2008, ISBN: 978-0521876346

[50] – Cristea Ana Daniela, Octavian Prostean, Thomas Muschalik și Tirian Ovidiu, *Distributed system for access control to physical resources based on qualifications*, WSEAS 2010, SEPADS'10 (SOFTWARE ENGINEERING, PARALLEL and DISTRIBUTED SYSTEMS) International Conference, University of Cambridge, Cambridge, UK, Februarie 20-22, 2010

[51] – Cristea Ana Daniela, Octavian Prostean, Muschalik Thomas și Tirian Ovidiu, The advantages of using SAP NetWeaver platform to implement a multidisciplinary project, ICCC-CONTI 2010, 27-29 Mai, Timișoara România

[52] – Ghosh, S: DISTRIBUTED SYSTEMS, Editura Chapman & Hall/CRC 2006, ISBN: 978-1584885641

[53] - Andrew S. Tanenbaum și Maarten van Steen, Distributed systems principle and paradigms, Practice Hall, New Jersy, ISBN 0-13-088893-1

[54] - Toma Leonida Dragomir, Elemente de teoria sistemelor, Editura Politehnica Timișoara 2004, ISBN 973-625-182-9

[55] – Tirian Ovidiu, Octavian Prostean, Rusu-Anghel Stela, Pinca-Bretotean Camelia și Cristea Ana Daniela, *Fuzzy system for implementing the cracks control during the continuous casting*, The 3rd European DAAAM International Young Researchers´ and Scientists´ Conference 25-28th November 2009, Vienna, Austria, ISBN 978-3-901509-70-4, ISSN 1726-9679

[56] – Tirian Ovidiu, Camelia Bretotean Pinca, Ana Daniela Cristea și Marcel Topor, *Research on the elimination of cracks in continuous casting plan using fuzzy logic*, WSEAS, Recent Advances in Circuits, Systems, Electronics, Control and Signal Processing, ISBN 978-960-474-139-7, ISSN 1790-5117, pagini 273-278 (conferință ISI Proceeding)

[57] - Loren Heilig, Steffen Karch, Oliver Böttcher, Christophe Mutzig, Jan Weber, Roland Pfennig, SAP NetWeaver: The Official Guide, ISBN 978-1-59229-193-9, 2008

[58] - Nancy Muir, Ian Kimbell , Discover SAP, SAP Press, 2008, ISBN 978-1-59229-320-9

[59] – Seitul oficial SAP help SDN: http://www.sdn.sap.com/irj/sdn

[60] - Karl Kessler, Peter Tillert și Panayot Dobrikov, Java Programming with the SAP Web Application Server, SAP Press, 2005, ISBN: 1-59229-020-5

[61] - Bertram Ganz, Jochen Gürtler și Timo Lakner, Maximizing Web Dynpro for Java, Editura SAP Press 2005, ISBN: 1-59229-077-9

[62] – Chris Whealy, Inside Web Dynpro for Java, Editura SAP Press 2007, ISBN: 978-3-89842-092-5

[63] - Gartner Group, Magic Quadrant, http://www.gartner.com

[64] - Sylvia Chaudoir, Mastering SAP ERP HCM, Organization Management, SAP Press 2009, ISBN 978-1-59229-208-0

[65] - Ewald Brochhausen, Jürgen Kielisch, Jürgen Scherring și Jeans Staeck, mySAP HR- Technical Principles and Programming, SAP Press 2005, ISBN 978-1-59229-055-0

[66] - Richard Haßmann, Christian Krämer și Jens Richter, Personalplanung und -entwicklung mit SAP ERP HCM, SAP Press 2009, ISBN 978-3836211222

[67] - Prashanth Padmanabhan, Christian Hochwarth, Sharon Wolf Newton, Sankara Narayanan Bharathan și Manoj Parthasarathy, SAP Enterprise Learning, SAP Press 2009, ISBN: 978-1-59229-269-1

[68] - Hans-Jürgen Figaj, Richard Haßmann și Anja Junold, HR Reporting with SAP, SAP Press 2008, ISBN: 978-1-59229-172-4

[69] - Stephan Kaleske, Praxishandbuch SAP Query-Reporting, SAP Press 2010, ISBN 978-3-8362-1433-9

[70] - Cristea Ana Daniela, Octavian Prostean, Mushalik Thomas și Ovidiu Tirian, Using SAP NetWeaver To implement a new authorization concept baset on qualifications and physical connection through RFID, The 3rd European DAAAM International Young Researchers´ and Scientists´ Conference 25-28th November 2009, Vienna, Austria, ISBN 978-3-901509-70-4, ISSN 1726-9679, paginile 1617-1619

[71] – Michael Hernandez, Proiectarea bazelor de date, Editura Teora 2003, ISBN: 973-20-0892-X

[72] - Cristea Ana Daniela, ABAP Dictionary used in Web Dynpro application, Jurnal of Engineering annals of Faculty engineering Hunedoara TOME VI 2008, Fascicula 3, ISSN 1584 – 2673, Pag 222 – 227

[73] - Karl-Heinz Kühnhauser, Discover ABAP, SAP Press 2008, ISBN 978-3-8362-1218-2

[74] – Ulrich Gellert și Ana Daniela Cristea, Praxishandbuch Web Dynpro ABAP, Editura Springer Berlin 2010, ISBN: 978-3-642-11386-4

[75] - Horst Keller și Sascha Kruger, ABAP Objects ABAP Programming in SAP NetWeaver, SAP NetWeaver Essentials, Editura SAP Press2006, ISBN: 978-1-59229-049-9

[76] - Horst Keller, The Official ABAP Reference volumul 1, Editura SAP Press 2005, ISBN: 978-1-59229-039-0

[77] - Horst Keller, The Official ABAP Reference volumul 2, Editura SAP Press 2005, ISBN: 978-1-59229-039-0

[78] - Andreas Blumenthal, Horst Keller, ABAP – Fortgeschrittene Techniken und Tools, Editura SAP Press 2005, ISBN: 978-3-89842-522-3

[79] – Christian Assig, Aldo Hermann Fobbe and Arno Niemietz, Object Services in ABAP, SAP Press 2009, ISBN 978-3-8362-1404-9

[80] – Rich Heilman și Thomas Jung, Next generation ABAP Development, Editura SAP Press 2007, ISBN: 978-1-59229-139-7

[81] - Cristea Ana Daniela, Adela Diana Berdie și Osaci Mihaela, Working with ABAP Persistent data, International Symposium "INTERDISCIPLINY REGIONAL RESEARCH", Hunedoara 23 - 24 Aprilie 2009, ISIRR 2009.

[82] - Andreas Wiegenstein, Markus Schumager, Sebastian Schinzel and Frederik Weidemann, Sichere ABAP Programmierung, SAP Press 2009, ISBN 978-3-8362-1357-8

[83] - Marcus Banner, Halil-Cem Gürsoy, Heinzpeter Klein, Mastering SAP NetWeaver XI—Programming, SAP Press 2007, ISBN 978-1-59229-140-3

[84] - Cristea Ana Daniela și Adela Diana Berdie, Error handling and messages with application Server ABAP, International Symposium "INTERDISCIPLINY REGIONAL RESEARCH", Hunedoara 23 - 24 Aprilie 2009, ISIRR 2009.

[85] - Ulli Hoffmann, Web Dynpro for ABAP, Editura SAP Press2006, ISBN: 1-59229-078-7

[86] - Dominik Ofenloch, Roland Schwaiger, Getting Started with Web Dynpro ABAP, Editura SAP Press 2010, ISBN: 978-1-59229-311-7

[87] - Cristea Ana Daniela, Adela Diana Berdie, Osaci Mihaela și Chirtoc Vasile Daniel, The Advantages of using Mind Map for learning Web Dynpro, published online la Computer Applications in Engineering Education în data de 19 Feb 2009, disponibil online la: http://www3.interscience.wiley.com/journal/122210203/abstract?CRETRY=1&SRETRY=0

[88] – Cristea Ana Daniela, Web Dynpro ABAP Presentation through a Mind Map, Jurnal of Engineering annals of Faculty engineering Hunedoara TOME VI, year 2008, Fascicula 3, ISSN 1584 – 2673, Pag 194 – 199

[89] - Cristea Ana Daniela, Berdie adela și Osaci Mihaela, Application componentization and component usages with Web Dynpro ABAP and Web Dynpro Java, SCIENTIFIC BULLETIN of "Politehnica" University of Timisoara, ROMANIA, Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE, Fasc. 1, 2010

[90] - Cristea Ana Daniela, Berdie Adela și Osaci Mihaela, The use and importance of Hook Methods in Web Dynpro ABAP and Web Dynpro Java, SCIENTIFIC BULLETIN of "Politehnica" University of Timisoara, ROMANIA, Transactions on

AUTOMATIC CONTROL and COMPUTER SCIENCE, Vol. 54 (68), Fasc. 3, 2009, ISSN 1224-600X

[91] - Cristea Ana Daniela și Octavian Prostean, Dynamic programming with Web Dynpro ABAP, 5th International Symposium on Applied Computational Intelligence and Informatics, May 28–29, 2009 – Timişoara, Romania

[92] - Cristea Ana Daniela, Adela Berdie și Mihaela Osaci, User interfaces with Web Dynpro ABAP and Web Dynpro Java, The Knowledge-based organization, The 14th International Conference, Technical Sciences Computer Science, Modeling & Simulation and E-Learning Technologies, Sibiu 27-29 NOVEMBER 2008, Pag. 157 – 164

[93] – Jurgen Hauser, Andreas Deutesfeld, Stephan Rehmann, thomas Szucs și Philipp Thun, SAP Interactive Forms by Adobe, SAP Press 2009, ISBN: 978-1-59229-254-7

[94] - Ulrich Gellert și Ana Daniela Cristea, Web Dynpro ABAP for Practitioners, Editura Springer Berlin 2010, ISBN: 978-3-642-11384-0

[95] - Ethan Cerami, Web Services Essentials: Distributed Applications with XML-RPC, Soap, UDDI and Wsdl, Editura O'Reilly 2002, ISBN: 978-8173663390

[96] - Frank Coyle, XML, Web Services and the Data Revolution, Editura Addison Wesley 2002, ISBN: 0-201-77641-3

[97] – Sas Jacobs, Beginning with XML with DOM and Ajax, Editura Apress 2006, ISBN 978-1-59059-676-0

[98] – Eric van der Vlist, XML schema, Editura O'Relly 2002, ISBN 0-596-00252-1

[99] – Steven Holzner, Sams Teach Yourself XML in 21 Days, Editura Sams 2003, ISBN 0-672-32576-4

[100] – Steven Holzner, inside XML, Editura New Riders Publishing 2000, ISBN 0-7357-1020-1

[101] - Martin Raepple, The Developer's Guide to SAP NetWeaver Security, Editura SAP Press 2007, ISBN: 978-1-59229-101-4

[102] - Valentin Niculescu, Katharina Klappert and Helmut Krcmar, SAP NetWeaver Portal, SAP Press 2008, ISBN 978-1-59229-145-8

[103] - Marcus Banner, Berthold Latka, Roland Schroth, Michael Spee, Praxishandbuch SAP NetWeaver Portal, Editura SAP Press 2008, ISBN: 978-3-8362-1077-5

[104] - IBM Business Consulting Services, SAP Authorization System, SAP Press 2003, ISBN 978-1-59229-016-1

[105] - Horst Keller and Wolf Hagen Thümmel, Official ABAP Programming Guidelines, SAP Press 2009, ISBN 978-1-59229-290-5

[106] – Cristea Ana Daniela, Messages internationalization with Web Dynpro ABAP, publicat online la SDN, 01 Mai 2009. A fost prezent pe prima pagina a topicului Web Dynpro ABAP, disponibil online la: http://www.sdn.sap.com/irj/sdn

[107] - Cristea Ana Daniela, Multilanguage Qualification Catalogue used in a WebDynpro Application, publicat online la SDN, 01 Septembrie 2009, disponibil online la: http://www.sdn.sap.com/irj/sdn

[108] – SHUYAN ZHAO, RALPH KRICKE șI ROLF-RAINER GRIGAT, TUNIR: A Multi-Modal Database for Person Authentication under Near Infrared Illumination, Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation, Corfu Island, Greece, February 16-19, 2007

[109] – J. Heaney, D. Hybertson, A. Reedy, S. Chapin, T. Bollinger, D. Williams și M. Kirwan, Information Assurance for Entreprise Engineering, in proceedings la PloP, Monticello, Illinois, 8-12 septembrie 2002

[110] - DJAMEL SAIGAA, N. BENOUDJIT, K. BENMAHAMED și S LELANDAIS Face Authentication Using Enhanced Fisher linear discriminant Model, 4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, Noiembrie 17-19, 2005 (pp155-160)

[111] - SANJAY R. GANORKAR, Iris Recognition: An Emerging Biometric Technology, Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation, Corfu Island, Grecia, Februarie 16-19, 2007

[112] - Seongan Lim și Ikkwon Yie, Probabilistic privacy leakage from Challenge-Response RFID authentication protocols, Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications, Athena, Grecia, August 24-26, 2007

[113] - Richard E. Smith, Authentication:From Passwords to Public Keys, Editua Addison-Wesley 2001, ISBN: 9780201615999

[114] - L. Y. Por, X. T. Lim și F. Kianoush, Background Pass-Go (BPG), a New Approach for GPS, 12th WSEAS International Conference on COMPUTERS, Heraklion, Greece, July 23-25, 2008

[115] – IEEE technical policy commitee, Developing National Policies on the Deployment of Radio Frequency Identification (RFID) Technology, Washington D. C., 23 aprilie 2009

[116] - Harold G. Clampitt, RFID Certification Textbook, Editura American RFID Solutions, 2007, ISBN: 978-0979428500

[117] - VALDIS PORNIEKS și EGILS GINTERS, Security problems of RFID authentication protocols, 6th WSEAS International Conference on SYSTEM SCIENCE and SIMULATION in ENGINEERING, Venice, Italy, November 21-23, 2007

[118] - US Dept. of Health Education and Welfare. Secretary's Advisory Committee on Automated Personal Data Systems, Records Computers and the Rights of Citizens, 1973.

[119] - Simson L. Garfinkel. Adopting Fair Information Practices in Low-Cost RFID Systems, în Ubiquitious Computing, September 2002 Privacy Workshop

[120] – S. Srinivasan, Akshai Aggarwal și Anup Kumar, RFID Security and Privacy Concerns, Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, Decembrie 16-18, 2005 (pp69-74)

[121] - Tobias Götz, SAP-Logistikprozesse mit RFID und Barcodes, SAP Press 2010, ISBN 978-3-8362-1382-0

[122] – C. T. Jones, Step7 in 7 steps, A practical guide to implementing S7-300/S7-400 Programmable Logic Controller, Editura Patrick-Turner Publishing 2006, ISBN: 978-1889101033

[123] – J. Mueller, Controlling with SIMATIC: Practice Book for SIMATIC S7 and SIMATIC PCS7 Control Systems, Editura Wiley-VCH 2005, ISBN: 978-3895782558,

[124] – Hans Berger, Automating with SIMATIC: integrated automation with SIMATIC S7-300/400, Editura Publicis Corporate Publishing 2004, ISBN: 978-3895782237

[125] - Karl Heinz John șI Michael Tiegelkamp, IEC 61131-3: Programming Industrial Automation Systems, Concepts and Programming languages reqierements for programming systems, Aid to Decision-Making Tools, Editura Springer 2001, ISBN 3-540-67752-6

[126] – SCHWARZ M.H. șI BOERCSOEK J., OPC for Process Maintenance, 6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, Tenerife, Spain, December 14-16, 2007

[127] - Wolfgang Mahnke, Stefan-Helmut Leitner și Matthias Damm, OPC Unified Architecture, Editura Springer 2009, ISBN: 978-3540688983

[128] - Software OPC:
http://www.softwaretoolbox.com/Prod_Services/DotNet/dotnet.asp

[129] - ABDALLAH.R.AL-ZYOUD și MAZOUZ.A.SALAHAT, USING PROGRAMMABLE LOGIC CONTROLLERS (PLCs), TO REALIZE DISTANCE PROTECTION, Proceedings of the 5th WSEAS Int. Conf. on Power Systems and

Electromagnetic Compatibility, Corfu, Greece, August 23-25, 2005 (pp325-328)

[130] – Siemens AG, SIMATIC S7-300 Programmable Controller, System Manual, 2004

[131] - Jiri Koziorek și Zdenek Slanina, Control System for the Prototype of Hydrogen Powered Car, 2005 WSEAS Int. Conf. on DYNAMICAL SYSTEMS and CONTROL, Venice, Italy, November 2-4, 2005 (pp369-373)

[132] – P. Ferrari, A. Flammini, D. Marioli, A.Taroni, An experimental approach to estimate real-time characteristic of PROFINET IO versus PROFIBUS DP V2, Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp104-111)

[133] – Internet Society ISOC, http://www.isoc.org

[134] – Introducere în ODP Reference Model, 2/14/96, Kazi Farooqui, Luigi Logrippo și Jan de Meer, The ISO Reference Model for Open Distributed Processing - An Introduction, disponibil online la http://www.enterprise-architecture.info/

[135] - Organizația patternurilor pentru securitate, adresa web a organizației: http://www.securitypatterns.org/patterns.html