

II 39.529

**UNIVERSITATEA “DUNAREA DE JOS” DIN GALAȚI
FACULTATEA DE AUTOMATICĂ, CALCULATOARE,
INGINERIE ELECTRICĂ ȘI ELECTRONICĂ CATEDRA DE
AUTOMATICĂ ȘI INGINERIE ELECTRICĂ**



ing. Iulian Nicușor ARAMĂ

**APLICAȚII ALE SISTEMELOR
MULTIAGENT ÎN DISTRIBUȚIA DE
ENERGIE ELECTRICĂ**

REZUMAT TEZA DOCTORAT

Conducator științific: prof. dr. ing. Viorel MÎNZU

IANUARIE 2011

39.529

ROMANIA
MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI
UNIVERSITATEA DUNĂREA DE JOS DIN GALAȚI

Strada Domnească nr. 47, cod poștal 800008
Galați, România
E-mail: rectorat@ugal.ro



Tel: (+4) 0336-130.109; 0336-130.108; 336-130.104
Fax: (+4) 0236-461.353
www.ugal.ro

C 60.8/26.01.2011

Către

Universitatea "Dunărea de Jos" din Galați vă face cunoscut că în data de 28.01.2011, ora 12.00, în sala Senatului Universității "Dunărea de Jos" din Galați, va avea loc susținerea publică a tezei de doctorat intitulată: "Aplicații ale sistemelor multi-agent în distribuția de energie electrică", elaborată de domnul/doamna ing. ARAMĂ IULIAN-NICUȘOR, în vederea conferirii titlului științific de doctor în Domeniul de doctorat - Automatică.

Comisia de doctorat are următoarea componență :

Presedinte:

Prof.dr.ing. Dorel AIORDĂCHIOAIE
Decan - Facultatea de Automatică, Calculatoare, Inginerie Electrică și Electronică
Universitatea "Dunărea de Jos" din Galați

Conducător de doctorat:

Prof.dr.ing. Viorel MÎNZU
Rector-Universitatea "Dunărea de Jos" din Galați

Referent 1:

Prof.dr.ing. Aurelian STĂNESCU
Universitatea POLITEHNICA din București

Referent 2:

Prof.dr.ing. Sergiu ILIESCU
Universitatea POLITEHNICA din București

Referent 3:

Prof.dr.ing. Sergiu CARAMAN
Universitatea "Dunărea de Jos" din Galați

Cu această ocazie vă transmitem rezumatul tezei de doctorat și vă invităm să participați la susținerea publică. În cazul în care doriți să faceți eventuale aprecieri sau observații asupra conținutului lucrării, vă rugăm să le transmiteți în scris pe adresa Universității, str. Domnească nr. 47, 800008 - Galați, Fax - 0236 / 461353.

RECTOR,
Prof.dr.ing. Viorel MÎNZU

SECRETAR DOCTORAT,
Ing. Luiza AXINTE

264 950
SCHEDE
SECRETARIAT

Cuprins

Cuprins	3
Introducere	5
Lista abrevieri	9
Capitolul 1	12
Sisteme autonome de control	12
1.1. Arhitectura sistemelor autonome de control	13
1.1.1. Funcțiile sistemului autonom de control	13
1.1.2. Structura sistemelor autonome de control	15
1.1.3. Sisteme autonome de control în rețelele de distribuție a energiei electrice	16
1.2. Implementarea sistemelor autonome de control	18
1.2.1. Cerințe privind arhitectura	18
1.2.2. Aspecte legate de comunicația de date	20
1.2.3. Modele standard de date	21
Capitolul 2	22
Sisteme multiagent	22
2.1. Modele de negociere multiagent în rețelele de distribuție a energiei electrice	22
2.2. Importanța, rolul și caracteristicile sistemelor multiagent	23
2.2.1. Utilitatea sistemelor multiagent	23
2.2.2. Caracteristici	23
2.3. Teoria negocierii și agenții inteligenți	23
2.4. Sistemul de negociere multiagent	25
2.4.1. Implementarea sistemului multiagent	25
2.4.2. Protocolul de negociere	26
2.5. Limbaje de comunicare inter-agent	27
2.5.1. Knowledge Query and Manipulation Language – KQML	27
2.5.2. Knowledge Interchange Format – KIF	28
2.5.3. Coordination Language – COOL	29
2.6. Sisteme blackboard	30
Capitolul 3	32
Sistem multiagent cu autonomie completa pentru restabilirea alimentării în rețelele de distribuție a energiei electrice	32
3.1. Problema restabilirii rețelei de distribuție a electricității în urma unei avarii	32
3.2. Modelul matematic privind restabilirea rețelei de distribuție a electricității	33
3.3. Arhitectura sistemului multiagent cu autonomie completa	34
3.4. Procesul de negociere	35
Capitolul 4	37
Implementarea unui sistem multiagent cu autonomie completa pentru restabilirea alimentării utilizând mediul de dezvoltare JACK	37
4.1. Implementarea sistemelor multiagent utilizând platforma JACK	37
4.2. Proiectarea și implementarea sistemului multiagent cu autonomie completa	38
4.2.1. Definirea agenților	38
4.2.2. Definirea evenimentelor și a mesajelor asociate acestora	39
4.2.3. Definirea planurilor de tratare a evenimentelor	40
4.3. Concluzie	42
Capitolul 5	43
Sistem multiagent cu autonomie supervizată pentru restabilirea alimentării cu energie electrică	43
5.1. Modurile de operare ale nodurilor rețelei de distribuție a energiei electrice	43

5.1.1 Auto-operarea	44
5.1.2 Operarea ierarhizată	45
5.1.3 Operarea arbitrată	46
5.2. Organizarea aplicației SMARA	47
5.3. Implementarea aplicației SMARA	49
5.4. Testarea aplicației SMARA	52
Capitolul 6.....	53
Direcții viitoare de cercetare pentru RDEE	53
6.1. Provocările sistemului de distribuție a energiei electrice	53
6.2. Rețele virtuale pentru distribuția energiei electrice	53
Concluzii	55
Referințe bibliografice	58

Introducere

În domeniul distribuției energiei electrice, sistemele automate de control au fost implementate de multă vreme, nu atât din considerente de eficiență economică, așa cum este cazul altor domenii, cât din considerente privind aspectele tehnice. Viteza de desfășurare a fenomenelor electro-energetice este mai mare decât viteza de percepție biologică și reacție umană, astfel încât sarcina operatorului uman a trebuit să fie preluată de sisteme automate de control.

Diferența fundamentală dintre vechile sisteme automate de control din domeniul distribuției electricității și cele noi este aceea că primele foloseau **date** obținute de la transductoare și senzori, pe când cele actuale folosesc **baze de informații**, ceea ce a ridicat nivelul de "inteligentă" al sistemelor automate de control, practic independent de nivelul tehnologic. Astfel, se pot concepe, cu tehnologii mai vechi, sisteme automate de control complexe, prin acest surplus de inteligentă. Sistemele de conducere mai vechi au un comportament "reflex", fiind "dedicate" anumitor sarcini, pe când cele noi pot utiliza fluxurile informaționale și au un comportament "inteligent". Ele sunt concepute pentru căutarea unei soluții prompte în vederea lichidării unei situații de criză sau o adaptare mai bună.

Din perspectiva acestor sisteme de conducere inteligentă, prezenta lucrare își propune să abordeze o problemă importantă din cadrul conducerii sistemelor de distribuție a energiei electrice și anume *problema restabilirii* acestui sistem în urma producerii unei avarii. Într-o astfel de situație, sistemele de protecție decuplează anumite zone ale rețelei de distribuție a energiei electrice (RDEE) și anumiți consumatori nu mai primesc energie electrică. Rezolvarea acestei probleme înseamnă reconfigurarea rețelei a.î. zonele fără energie electrică să fie conectate la zonele neafectate prin intermediul întrerupătoarelor de linie, asigurându-se în același timp necesarul de putere pentru zonele nou conectate. Pe de altă parte, căutarea de linii care pot furniza energie electrică, luarea deciziei de conectare și acționarea întrerupătoarelor de linie necesită un anumit timp. Acesta trebuie să fie cât mai scurt pentru a asigura un indicator de calitate esențial în distribuția de energie electrică: continuitatea.

În această lucrare, modul de abordare propus în soluționarea problemei restabilirii este cel al *sistemelor multiagent*. Motivatia este legată de natura problemei, care este o problemă de căutare într-o rețea de obicei complexă și care trebuie să se facă într-un timp scurt. De aici necesitatea de a recurge la metodele *inteligentei artificiale distribuite*. Mai mulți agenți de

cautare, distribuiti geografic în RDEE, pot cauta soluții de reenergizare a unor linii, în zonele vecine și pot negocia direct cu acestea pentru o anumita cerință de putere. Se poate găsi astfel o soluție care, chiar dacă nu este optimă, este fezabilă și se obține relativ repede. Implementarea unei astfel de soluții duce la realizarea unor entități soft autonome, agenții inteligenți, ce comunică între ele și care utilizează o bază de informații distribuită. Constatăm că soluția implementată este un *sistem autonom de control*, tocmai pentru că este formată din entități autonome distribuite geografic și care utilizează o bază de informații, pentru a rezolva un consemn general, acela de a reenergiza zonele avariate.

Iată de ce, în Capitolul 1, se prezintă un "state of the art" al *sistemelor autonome de control* (comanda) -SAC, plecând de la ideea că o "comandă" de restabilire automată a RDEE avariata poate fi furnizată de un dispecer central, dar trebuie "realizată" de entități autonome situate la distanță, acolo unde sunt și informația, și modalitățile practice de soluționare. În secțiunea consacrată *arhitecturii* sistemelor autonome de control, aspectul esențial este faptul că fiecare entitate autonomă are trei nivele funcționale: *organizare/management*, *coordonare* și *execuție*. Plecând de la legătura evidentă dintre un sistem autonom și un agent inteligent, putem constata că un sistem cu mai multe entități autonome circumscrise aceluiași obiectiv și sistemele multiagent (SMA) este o legătură imediată. De aici, pleacă și ideea de bază din această lucrare, că sistemele autonome de control pot fi implementate prin SMA. Autonomia în execuție nu împiedică asupra existenței unor legături ierarhice la nivelul coordonării. Legătura ierarhică dintre entitățile autonome este foarte importantă, acest fapt punându-se pe seama asupra tipului de SMA ce realizează implementarea.

În secțiunea 1.1.3 structura generală a unui SAC este repercutată pe cazul rețelei de distribuție a energiei electrice. Aspectele practice de infrastructură a implementărilor SAC fac obiectul secțiunii 1.2 a acestui capitol. Acesta se referă la infrastructura de comunicații, modele de date și baza de informații.

Cea de-a doua axă pe care se dezvoltă prezenta lucrare este cea a inteligenței artificiale distribuite care aici îmbracă forma *Sistemelor MultiAgent* (SMA). Modul practic în care se implementează un SAC cu mai multe entități autonome poate fi un sistem multiagent, tot așa cum un regulator poate fi implementat printr-un program ce rulează pe un micro-controller. Capitolul 2 face o prezentare a Sistemelor Multiagent, presupunând că cititorul prezentei lucrări este familiarizat deja cu noțiunea de agent inteligent. Sunt trecute în revistă caracteristicile definitorii ale unui SMA, insistându-se mai mult pe modul de realizare a

negocierii între agenți și pe comunicarea inter-agent. Sunt trecute în revista și principalele limbaje de comunicare, KQML, KIF și COOL, precum și sistemele de tip blackboard.

Prezenta lucrare propune două soluții la problema restabilirii în RDEE. O primă soluție propune un SAC cu o structură ierarhică foarte "plată" și cu agenți omogeni complet autonomi în funcționare. O a doua soluție introduce o ierarhie mai stratificată în timp, atunci când se adoptă diferite moduri de operare ale RDEE.

Capitolul 3 intitulat "Sistem multiagent cu autonomie completă pentru restabilirea alimentării în rețelele de distribuție a energiei electrice", după ce în prima sa secțiune formulează problema restabilirii după avarie în RDEE, iar în cea de a doua furnizează modelul sau matematic, propune ca soluție structura unui SMA cu autonomie completă, după o idee prezentată în [94]. Agenții au o arhitectură BDI (convingere-dorință-intenție). Sunt trecute în revista, pe baza unui studiu de caz, caracteristicile procesului de negociere dintre agenți și se face o analiză prin simulare calitativă a dinamicii SMA în găsirea unei soluții. Realizarea practică a sistemului multiagent face obiectul Capitolului 4, intitulat "Implementarea unui sistem multiagent cu autonomie completă pentru restabilirea alimentării utilizând mediul de dezvoltare JACK". Contribuția nu este o simplă programare într-un sistem dat, ci se referă la crearea în universul problemei a unor proceduri de comunicare, negociere și de un mod de implementare a arhitecturii BDI care să asigure ca procesele iterative desfășurate de agenți duc la o soluție a problemei în timp util. Aici, un aspect important este definirea evenimentelor, mesajelor inter-agent și a planurilor de tratare a evenimentelor de către agenți. Într-un mod plastic exprimat, agenții sunt "încarcați" cu proceduri și "inteligenta", astfel încât, o dată lăsați liberi, să fie capabili să interacționeze, să modifice universul lor și propriul "interior", până când conturează o soluție.

În capitolul 5, intitulat "Sistem multiagent cu autonomie supervizată pentru restabilirea alimentării cu energie electrică", se propune cea de-a doua soluție de SMA invocată mai sus și care se referă la un mod de operare dinamic în RDEE, care duce la ierarhizări între agenți. Se propun arhitectura și toate elementele de infrastructură realizate în Sistemul MultiAgent pentru Restabilirea Alimentării cu energie electrică - SMARA. De data aceasta, sistemul este implementat în JAVA și are o concepție originală fără a utiliza pachete specializate. S-a simțit nevoia unei noi soluții de tip SMA, pentru că, deși ierarhizarea "strică" din simplitatea elegantă a agenților omogeni și neierarhizați, în felul acesta s-a putut utiliza experiența dispeceratelor în rezolvarea problemei restabilirii și micșorarea timpului de obținere a soluției.

În Capitolul 6, sunt trecute în reviste doua directii de cercetare care ar putea duce la noi functionalitati ale aplicatiei SMARA si anume: trecerea de la o retea pasiva (structura fixa) la una activa (avand configuratie buclata si producatori distribuiti geografic), precum si implementarea retelelor virtuale de distributie a energiei electrice (SMART Grid).

In capitolul "Concluzii", sunt trecute în revista principalele contributii si publicatiile aferente activitatii de cercetare desfasurate de autor, în domeniul stiintific abordat în aceasta lucrare.

Lista abrevieri

Abrevieri din limba română:

A.A.R.	Anclanșarea Automată a Rezervei
DASf	Declanșare automată secvențială la depășirea frecvenței
DEN	Dispecerul Energetic Național
DT	Dispecer Teritorial
ELECTRE	acronimul francez pentru <i>ELimination Et Choix Traduisant la REalité</i> , "eliminare și alegere traducând realitatea"
M.T.	Medie Tensiune
P.T.-M.T./J.T.	Post de Trasformare din Medie Tensiune în Joasă Tensiune
R.A.T.	Regulator Automat de Tensiune
RDEE	Rețea de Distribuție a Energiei Electrice
SMARA	Sistem Multi-Agent pentru Restabilirea Alimentării cu energie electrică
S.T.110kV/MT	Stație de Transformare din 110 kV în Medie Tensiune
U.E.	Uniunea Europeană

Abrevieri din limba engleză:

ACS	Algorithms for Complex Shapes with certified topology and numeric Algoritmi numerici pentru Modelarea Complexă cu ajutorul topologiilor precise
API	Application Programming Interface Interfață de programare a aplicațiilor (informatice)
ATM	Asynchronous Transfer Mode Transfer asincron al datelor
BDI	<i>Beliefs, Desires, Intentions</i> Convingeri, Dorințe, Intenții
CASE	<i>Computer Aided Software Engineering</i> Soft pentru ingineria asistată de calculator
COOL	<i>Classroom Object Oriented Language</i> Limbaj orientat pe clase de obiecte
DFR	<i>Digital Fault Recorders</i>

	Agent pentru înregistrarea defectelor ireversibile (din rețelele de transport și distribuție a electricității)
DMS	<i>Distribution Management System</i> Sistem de gestionare a distribuției electricității
EMIS	Enterprise Management Information System Sistem informatic de gestionare a întreprinderii
EMS	<i>Energy Management System</i> Sistem de gestionare a energiei (în cadrul unei companii de producere, transport, distribuție sau a unui mari consumator de electricitate)
EMS-API	Energy Management System Application Programming Interface Interfață de programare a aplicațiilor (informatic) pentru sistemele de gestionare a energiei
FRI	Fault Recorder and Interpreter Agent de înregistrare și interpretare a defectelor
FRR	Fault Receiver and Recorder Agent de percepere și înregistrare a defectelor
FTP	File Transfer Protocol Protocol de transfer a fișierelor
GW	Gateway "Poartă de acces" într-o rețea de calculatoare
HTTP	Hyper Text Transport Protocol Protocol de transport a textelor
IEI	Identification of Events and Incidents Identificarea evenimentelor și incidentelor (în rețelele de transport și distribuție a electricității)
JESS	Java Expert System Shell Interfață-sistem expert folosind limbajul <i>Java</i>
KIF	Knowledge Interchange Format Limbaj (orientat) pe schimbul de cunoștințe între aplicații / programe diferite
KQML	Knowledge Query and Manipulation Language Limbaj de consultare și manipulare a cunoștințelor
IED	Intelligent Electronic Device Aparat electronic inteligent=unitate de calcul digitală dedicată entităților tehnologice precum celula din instalațiile electro-energetice

LAN	Local Area Network Rețea locală de calculatoare
MAS	Multi-Agent System Sistem multi-agent
MMS	Manufacturing Message Specification Elaborarea Mesajelor specifice (în cadrul unei aplicații informatice)
PEDA	Protection Engineering Diagnostic Agents Agenți de diagnoză în ingineria protecțiilor
PSO	Particle Swarm Optimization Optimizarea prin metoda roiului de albine
PVD	Protection Validation and Diagnosis Agent pentru diagnoza și validarea protecțiilor
RTU	Remote Terminal Unit Unitate Terminală la Distanță=unitate de calcul implementată în instalații ale procesului tehnologic
SCaDA	Supervisory Control and Data Acquisition Achiziție de date și control de supraveghere (a procesului tehnologic)
SCL	Substation Configuration Language Limbaj de configurare a stațiilor de transformare (destinat utilizării de către energeticienii fără pregătire informatică solidă în vederea informatizării procesului tehnologic din instalațiile energetice)
TP	<i>Telemetry Processor</i> Procesor de telemetrie (utilizat în SCaDA pentru procesarea semnalizărilor și măsurilor preluate de la RTUs)
TSP	<i>Travelling Salesman Problem</i> Problema comis-voiajorului
UML	<i>Unified Modelling Language</i> Limbaj unificat pentru modelarea (informatică) a proceselor tehnologice
WAN	<i>Wide Area Network</i> Rețea de calculatoare răspândită geografic

Capitolul 1

Sisteme autonome de control

Conceptul de sistem autonom provine din robotica. De exemplu, un robot poate găsi un drum favorabil până la o poziție țintă ce reprezintă obiectivul curent al conducerii. Acest obiectiv este transmis robotului ca și cerința curentă, iar acesta în mod autonom va găsi un drum și se va deplasa către poziția indicată.

În [6] este prezentat conceptul de sistem autonom adoptat și în această lucrare. Necesitatea acțiunii autonome este explicată în lucrarea menționată printr-un exemplu, în cadrul căruia un robot se joacă cu mingea pe Luna. Un proces de conducere convențională ar utiliza senzorii robotului pentru a detecta mingea și ar trimite semnalele de la senzori pe Pământ. Controlerul ar stabili acțiunile de mișcare ale robotului și ar trimite semnale corespunzătoare înapoi robotului pe Luna. Acesta ar executa mișcări în concordanță cu semnalele primite de pe Pământ. Din cauza distanței, întârzierile în propagarea semnalelor sunt atât de mari încât acțiunile de mișcare ale robotului nu se mai potrivesc cu situația actuală, care s-a schimbat între timp. Conducerea autonomă în acest caz ar însemna că robotul are predefinite cunoștințe despre posibilele sale acțiuni, cum ar fi acțiunea „prinde mingea”. Controlerul de pe Pământ va genera și va transmite numai consemne de nivel înalt cum ar fi „prinde mingea” fără să se preocupe de toate acțiunile de mișcare ale robotului. Robotul primește consemnul și-l traduce în acțiuni concrete și mai fine, ținând cont de situația actuală robot-minge, de genul „mergi înainte” sau „ridică bratul”. Robotul își va coordona mișcările, își va adapta mișcările în funcție de obstacole sau alți roboți. Acțiunile propriuzise de mișcare sunt îndeplinite în ultima instanță. Procedura de conducere autonomă distinge între fazele Organizare, Coordonare și Executie.

Sistemul autonom se caracterizează prin reacția inteligentă și flexibilă la schimbarea condițiilor de lucru și a cerințelor trimise de procesele înconjurătoare. Comportamentul unui sistem autonom este similar celui din tehnologia informației, unde sunt sisteme orientate pe obiecte sau bazate pe componente. O cerere sau o indicație a utilizatorului este trimisă direct către obiectul sau componenta de la care se așteaptă apoi reacția dorită. În cazul sistemelor autonome, cerința este direcționată către o componentă care va acționa sau răspunde corespunzător cu o informație. Cerința este procesată mai mult sau mai puțin autonom, în funcție de starea procesului și de condițiile înconjurătoare. În acest context, "autonom" înseamnă că acțiunea utilizatorului se rezumă la definirea țintei, scopului sau cererii sale,

calea de atingere fiind aleasă pe proprie răspundere de către sistem. Pe măsură ce crește "răspunderea sistemului", va crește și gradul de *autonomie* al acestuia.

Sistemele autonome sunt întâlnite în procesele complexe, acelea în cadrul cărora apar cereri flexibile și de înalt nivel. Aceasta situație este în legătură cu faptul că acțiunile trebuie implementate în pofida existenței unui minim de cunoștințe a priori sau chiar a incertitudinilor. De aici și faptul că trebuie adoptate tehnici de inteligență artificială sau computațională.

Dacă definirea țintelor sau cerințelor de către utilizator și acțiunile sistemului autonom servesc conducerii unui proces, atunci este vorba de un *sistem autonom de conducere(control)*. Dacă servesc pentru obținerea unor informații structurate la un nivel înalt, atunci este vorba de un *sistem autonom informațional*. De exemplu, o componentă a unui sistem complex se poate ocupa cu supervizarea unui proces sau a unui subsistem, dând ca rezultat o informație ce reprezintă diagnosticul acestuia, ce poate fi transmisă operatorului procesului respectiv sau altor componente autonome.

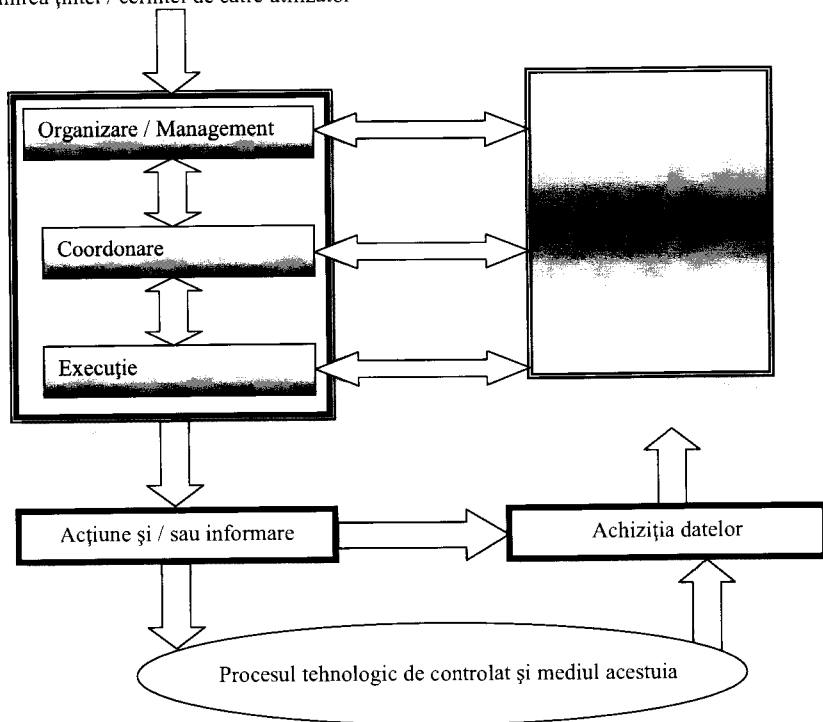
1.1. Arhitectura sistemelor autonome de control

Concepția arhitecturală are la bază teza incertitudinii condițiilor de lucru, ceea ce implică legătura strânsă cu domeniul *inteligenței artificiale* și *inteligenței calculatorului* [1]. Aceasta înseamnă că sistemul autonom de control trebuie să integreze abilitățile intrinseci inteligenței, adică învățarea din experiență, planificarea, precum și detectarea și identificarea erorilor, tocmai pentru a face față cererilor și indicațiilor utilizatorului.

1.1.1. Funcțiile sistemului autonom de control

Definirea funcțiilor unui sistem automat de control este o chestiune complexă și constituie rezultatul unui proces amplu de elaborare, dar aceste funcții pot fi grupate conform unei logici de funcționare. Aceasta este ilustrată în fig.nr.1 privind schema funcțională a unui sistem autonom. Funcțiile sunt definite pe trei nivele în funcție de atribute, adică "organizare/gestionare", "coordonare" și "execuție"[2], fiecare nivel având atribute clar definite.

Definirea țintei / cerinței de către utilizator

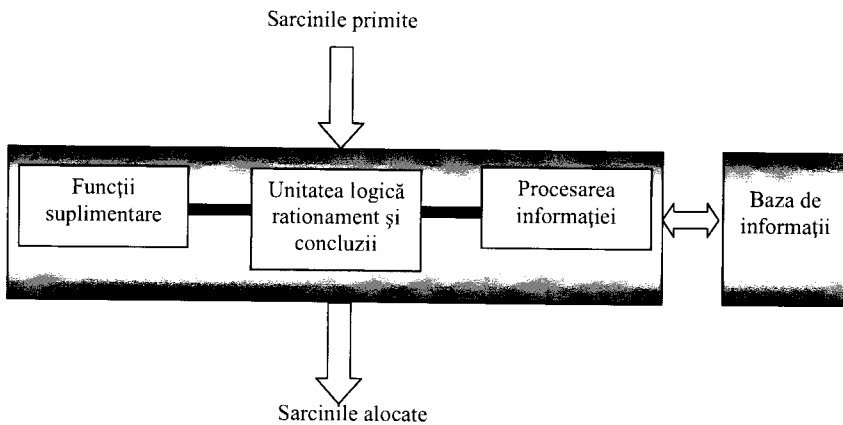


- Figura nr.1: schema funcțională a unui sistem autonom -

Sistemul funcționează pe baza definirii țăintelor sau scopurilor utilizatorului atunci când acesta își lansează cererea sau indicația. Acesta este compus din trei unități:

- *unitatea de prelucrare a informației*
- *unitatea logică concluzii/rationamente*
- *unitate pentru funcții suplimentare;*

În cadrul fiecărui nivel funcțional, unitatea logică concluzii/rationamente se comportă în mod similar unității aritmetice și logice a calculatorului obișnuit, unitate de conexiune la informație și procesarea acesteia se comportă în mod similar interfeței cu memoria externă a calculatorului obișnuit, iar unitatea funcțiilor suplimentare se comportă în mod similar bibliotecilor cu funcții ale sistemului de operare al calculatorului obișnuit.



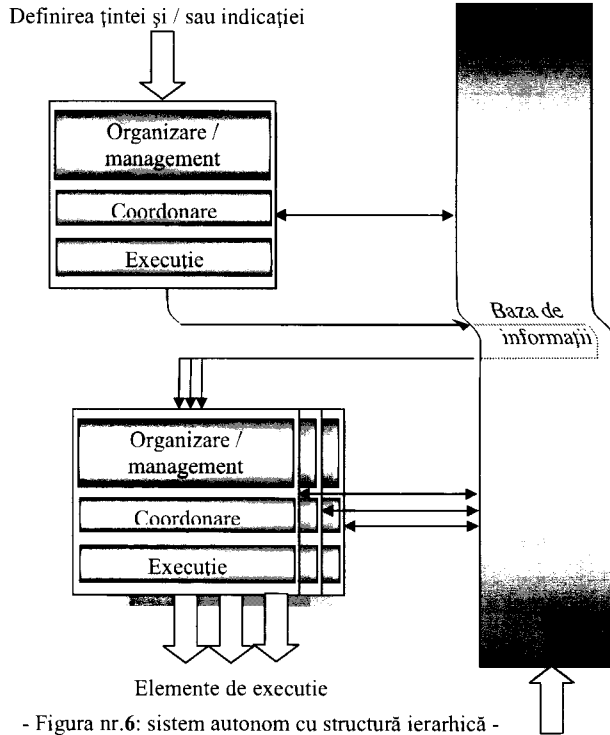
- Figura nr.2: schema unui nivel funcțional -

În realitate, funcționarea sistemului autonom este cu mult mai complexă, dar, în principiu, desfășurarea proceselor unui sistem autonom de control este aceea descrisă mai sus. Precizarea aceasta este importantă deoarece face comprehensibilă prezentarea nivelelor funcționale, definirea "gestionării", "coordonării" și "execuției" nefiind arbitrară, ci răspunzând unei logici funcționale deja verificate în domeniul calculatoarelor. Descrierea nivelelor funcționale se va face de jos în sus, pornind de la nivelul funcțional al "execuției", urmat de acela al "coordonării" și, în sfârșit, acela al "gestionării și organizării".

1.1.2. Structura sistemelor autonome de control

În cele de mai sus s-a descris o singură componentă autonomă deoarece s-a urmărit analiza funcționalității sistemelor autonome de control. În realitate, un sistem tehnic complex căruia i s-au alocat mai multe sarcini are mai multe componente autonome, fapt de care trebuie să se țină seama la analiza structurii sistemului autonom de control. Ca entitate acționând sub propria sa responsabilitate, un sistem autonom de control poate fi denumit *agent inteligent* ce acționează în funcție de condițiile mediului în care își desfășoară activitatea. Dacă în cadrul sistemului acționează simultan mai mulți agenți, atunci se poate spune că este vorba de un sistem *multi-agent*. Sistemele autonome sunt organizate într-o structură mai mult sau mai puțin ierarhică [3], [4]. Structura ierarhică din fig.nr.6 permite separarea funcțiilor și subfuncțiilor pe diferitele nivele ale structurii ierarhice. O componentă

autonomă de nivel superior poate apela simultan mai multe componente autonome de pe nivelul imediat inferior.



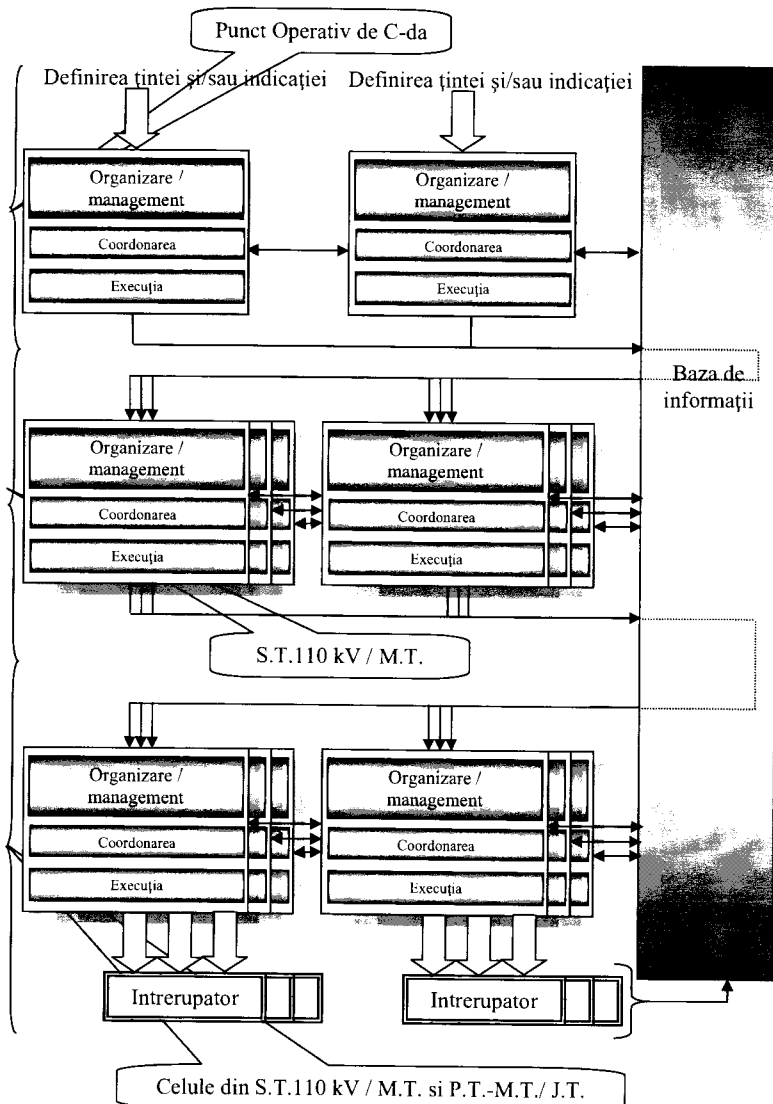
Componenta de pe nivelul superior operează în mod mai abstract, ieșirea sa fiind definirea sarcinilor pentru componentele de pe nivelul inferior.

1.1.3. Sisteme autonome de control în rețelele de distribuție a energiei electrice

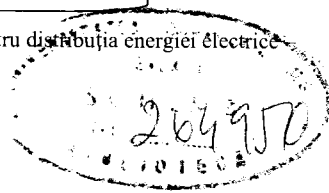
Sistemele autonome de control ale rețelelor de distribuție a energiei electrice sunt structurate ierarhic. Punctul Operativ de Comandă sau dispeceratul supraveghează și optimizează procesul tehnologic de distribuție a energiei electrice, aceasta constând în schimbarea stării aparatelor de comutație primară și a unor sarcini mai ușoare de reglaj. Acțiunile de comandă și de protecție, tocmai pentru că necesită un timp mult mai scurt de reacție, sunt efectuate în mod descentralizat prin implementarea circuitelor de protecție, AAR, RAT, etc.

Arhitectura operațională – vezi fig.nr.8.

Este nevoie de a trece actuala structură ierarhică a sistemelor de control a rețelelor de distribuție a energiei electrice la sistemul autonom de control distribuit la nivelul celulelor și P.T.-M.T./J.T., S.T.110 kV / M.T. și Punctului Operativ de Comandă.



- Figura nr.8: sistem autonom de control pentru distribuția energiei electrice



Arhitectura funcțională descrisă la paragraful 1.1.1 este transpusă în arhitectura operațională specifică sistemelor de control a rețelelor de distribuție a energiei electrice. Grupurile de componentă se referă la protecții, încadrarea tensiunii în bandă, circulații de puteri, fiabilitate, etc., fiecare componentă autonomă fiind dedicată unei anumite sarcini.

1.2.Implementarea sistemelor autonome de control

În această secțiune este prezentată arhitectura unui sistem autonom din punctul de vedere la Tehnologia Informatiei. Cu titlu de exemplu ce ne interesează în mod particular, prezentarea ce urmează consideră ca implementarea se face într-o rețea de distribuție a energiei electrice având deja un sistem tradițional de control.

1.2.1.Cerințe privind arhitectura

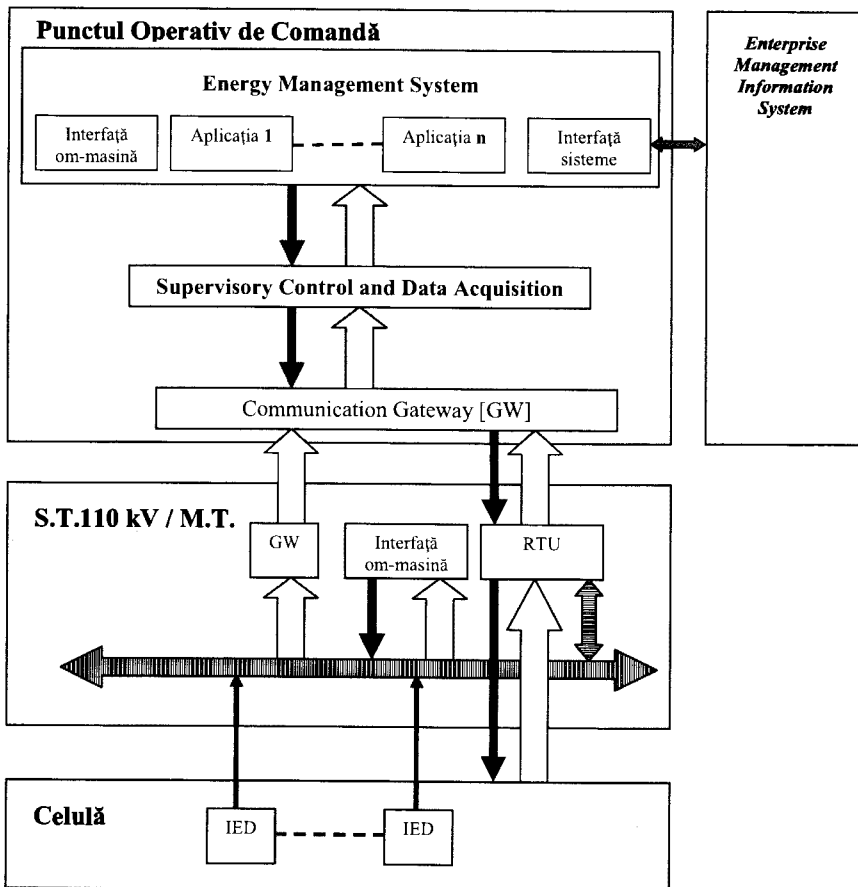
Arhitectura ce urmează a fi implementată are la bază logica relației dintre sistemul autonom de control și procesul tehnologic privind distribuția energiei electrice. Caracteristicile esențiale ale sistemelor autonome sunt:

- algoritmilor prin care se realizează prelucrarea inteligentă a informației și
- implementarea tehnică a sistemului autonom, adică *arhitectura implementării*.

Cerința de bază a arhitecturii implementării sistemului autonom este însăși cerința de bază a tehnologiei informației, accesul liber la date și schimbul liber de informație între componentele autonome. Tocmai pentru că cerința de bază a implementării sistemelor autonome este cea care stă la baza tehnologiei informației, rezultă de la sine necesitatea folosirii modelelor, protocoalelor și tehnologiilor de comunicație standard, numai această cerință permițând practic implementarea arhitecturii unui sistem autonom. În ceea ce privește distribuția energiei electrice, există câteva standarde privind automatizarea și tehnologia informației, standarde ce au fost expres implementate pentru promovarea soluțiilor optime în ceea ce privește relația cost-eficiență, motiv pentru care aceste standarde trebuie luate în considerare la implementarea sistemelor autonome de control. Actualele sisteme de control, cele tradiționale, au o arhitectură specifică – vezi fig.nr.9.

Arhitectura unui sistem informatic trebuie să asigure achiziționarea datelor în timp real, prelucrarea complexă a informației și auto-diagnoza, cerințe îndeplinite mai mult sau mai puțin de sistemele tradiționale de control, prea puternic ierarhizate. În schimb, sistemele autonome de control oferă servicii ce permit accesul larg și non-ierarhizat la informație,

schimbul de date în timp real între diversele componente la nivel de celulă și realocarea activă a funcțiilor către alte componente în caz de indisponibilitate.



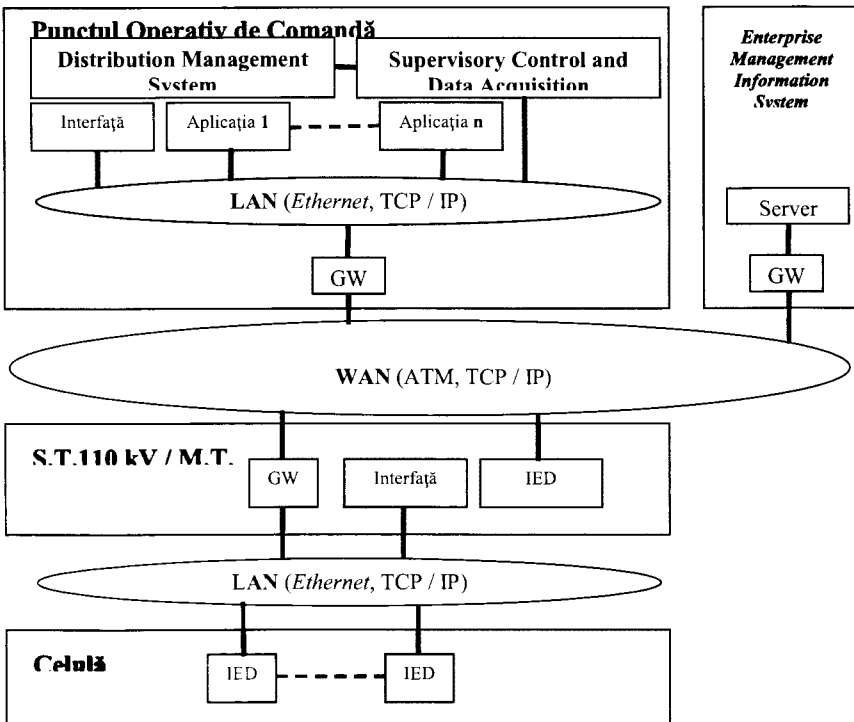
- Figura nr.9: Arhitectura sistemului tradițional de control -

Creșterea gradului de prelucrare locală a datelor permite sistemului autonom de control să reducă semnificativ traficul de informație între nivelele ierarhice, totodată permițând creșterea traficului de informație între componentele aceluiași nivel ierarhic. Toate acestea determină cerințe specifice privind implementarea arhitecturii sistemului autonom de control

1.2.2. Aspecte legate de comunicația de date

Din perspectiva arhitecturii implementării, structura fizică a unui sistem autonom de control este un sistem distribuit de calcul [7]. Sistemul autonom de control are o arhitectură de module software bazate pe componente cu stocare de date distribuita, aplicațiile fiind organizate în conformitate cu aspectele logice și funcționale ale procesului tehnologic, ceea ce este posibil datorită accesului liber la date și la stocarea datelor în locații independente.

Aspectele legate de topologie sunt prezentate în fig.10. Putem remarca faptul ca magistrala dintr-o S.T.110 kV / M.T. a fost înlocuită cu o LAN – *Local Area Network*, la acestea conectându-se direct IED, GWs și interfața om-mașină, comunicația cu Punctul Operativ de Comandă și cu S.T.110 kV / M.T. învecinate făcându-se prin intermediul unei WAN – *Wide Area Network*, în cadrul Punctului Operativ de Comandă implementându-se o altă LAN.



- Figura nr.10: noua arhitectură a comunicației -

În ceea ce privește protocolul comunicației, cerința esențială este garantarea comunicării datelor în timp real pe de o parte și eficiența transmiterii datelor ce nu implică timpul real. Al II-lea "strat" al modelului comunicației, acela al legăturii de date, poate fi implementat cu ajutorul ATM – *Asynchronous Transfer Mode*, acesta fiind un compromis între comunicația orientată spre conexiunile punct-cu-punct și comunicația orientată spre pachete [9]. În ciuda faptului că operează asincron și prin multiplexare, ATM poate fi folosit și pentru comunicații sincrone în cazul utilizării unor lungimi fixe ale celulelor de informație. Informația precizând conexiunea pentru prelevarea datelor intrinseci unei anumite celule este înscrisă în *header*-ul celulei.

1.2.3. Modele standard de date

Protocoalele de comunicație sunt modul "fizic" prin care se efectuează în timp real schimbul de date și de informație în cadrul sistemelor autonome de control. La rândul lor, datele trebuie astfel modelate încât să fructifice la maximum valențele comunicației, atât în ceea ce privește operarea în timp real, cât și în ceea ce privește efectuarea acelor sarcini ce nu implică timpul real, sarcini de genul configurării sistemului sau comunicării între aplicațiile sistemului. Practic, în majoritatea cazurilor se folosesc "straturile" inferioare ale OSI – vezi paragraful 1.2.2. În acest caz modelul standard al datelor este un vector conținând adresele semnalelor și de aceea nu există noțiunea de "aparat inteligent" la nivelul protocolului. Este vorba, deci, de o comunicație specifică schimbului anonim de date cu punctele identificate prin adresele de semnal deoarece aparatul receptor al semnalelor, cel mai adesea un RTU, nu cunoaște nici semnificația valorilor primite, nici care echipament fizic l-a emis; se știe doar că semnalul provine de la o anumită adresă. Aceasta poate fi acceptabil pentru sistemul tradițional de control, dar nu este de nici un folos sistemului autonom de control, de unde necesitatea modelării complexe a structurii datelor.

Standardul IEC 61850 marchează un moment important în depășirea situației descrise imediat mai sus. Deși în fond este un protocol de comunicație, IEC 61850 a fost conceput cu valențe de analiză a domeniului. A rezultat un model elaborat al domeniului, care conține și modelul de date. Acest model al domeniului se bazează pe conceptul de *nod logic* (LN), care, din punctul de vedere al sistemului autonom de control, trebuie privit ca funcția maxim-atomizată ce poate fi atribuită unui aparat inteligent din incinta S.T.110 kV / M.T., aparate de genul protecțiilor, AAR, etc.

Capitolul 2

Sisteme multiagent

Obiectivul acestui capitol este, în primul rând, de a justifica alegerea sistemelor multiagent pentru implementarea unui sistem autonom de comanda în rețelele de distribuție a RDEE. Necesitatea de reprezentare a entităților autonome și capacitatea acestora de a negocia sunt cheia alegerii facute.

În al doilea rând, sunt trecute în revista elemente esențiale din teoria sistemelor multiagent, legate de caracteristici, comunicare și negociere.

2.1 Modele de negociere multiagent în rețelele de distribuție a energiei electrice

Există o gamă largă de probleme de decizie în domeniul rețelelor de distribuție a energiei electrice, împărțite de obicei în trei categorii, probleme de decizie în domeniul planificării, exploatarea și întreținerea; în cadrul fiecăreia dintre cele trei categorii mai apar o serie de aspecte legate pe de o parte de natura deciziei și de orizontul de timp disponibil pentru elaborarea deciziei și executarea acțiunii.

Există probleme precum conducerea operativă a producției de electricitate, programarea aprovizionării cu combustibil, telecomenzi pentru prevenirea și / sau lichidarea avariilor, (re)porniri ale instalațiilor, etc., toate acestea având în comun următoarele aspecte:

- impactul deciziilor asupra funcționării rețelei de distribuție a energiei electrice este enorm, drept care se impune coordonarea informației între diferiții factori de decizie,
- stabilitatea funcționării se asigură numai prin alocarea unor mari resurse financiare,
- variabilele ce stau la baza elaborării deciziilor se caracterizează printr-un mare grad de incertitudine,
- gradul de incertitudine se reduce numai prin achiziția și prelucrarea informației,
- sursele informației sunt răspândite pe o mare arie geografică și
- sunt necesare aplicații complexe ce necesită un volum mare de calcule de înaltă complexitate.

2.2.Importanța, rolul și caracteristicile sistemelor multiagent

2.2.1.Utilitatea sistemelor multiagent

Sistemele multiagent își pot găsi utilitatea și în domeniul *rezolvării distribuite a problemelor*. Dacă domeniul unei probleme poate fi divizat în mai multe subprobleme de complexitate mai redusă, acestea pot fi ușor atribuite unor agenți care să le rezolve separat, în paralel, pentru ca apoi rezultatele parțiale să fie combinate pentru a forma soluția finală. În cazul în care problema are o natură distribuită, această abordare este mai comodă și strategia de rezolvare este mai ușor de înțeles și implementat. Din punctul de vedere al programatorului, pentru subproblemele rezultate pot fi produși algoritmi eficienți, în timp ce un algoritm centralizat ar fi prea complex sau ar putea chiar să nu reziste.

2.2.2.Caracteristici

Pentru sistemele multiagent au fost propuse numeroase taxonomii. Se consideră importante patru dimensiuni [64] ale inteligenței artificiale distribuite:

- granularitatea agenților – poate fi mare sau fină
- eterogenitatea cunoștințelor agenților - cunoștințele din sistem pot fi redundante
- metodele de control distribuit - agenții pot coopera sau pot fi în competiție; organizarea lor poate fi ierarhic sau orice firmă de control centralizat poate fi evitată
- posibilitățile de comunicare: comunicarea se poate realiza prin intermediul mesajelor transmise direct de la un agent la altul sau prin intermediul unei memorii comune –

2.3.Teoria negocierii și agenții inteligenți

Atribuirea abilităților sociale către agenții inteligenți pentru metamorfozarea acestora în sisteme multiagent, abilități de genul "negocierii", a început încă din anii '80, dar comportament similar celui uman nu s-a obținut decât recent. A fost necesară fundamentarea "teoriei negocierii" pe de o parte și fundamentarea atribuirii abilităților sociale către agenții inteligenți pe de altă parte. Nu-i mai puțin adevărat că nici nu au existat cerințe din partea proceselor tehnologice până la dezagregarea marilor companii de electricitate, dereglementarea pieței de electricitate și apariția intereselor concurențiale.

Negocierea este forma fundamentală a interacțiunii umane, fapt ce poate fi constatat în toate domeniile de activitate. A fost studiată și fundamentată abia în zilele noastre [39]. Cele de mai jos abordează teoria negocierii din perspectiva utilizării acesteia pentru agenții inteligenți. Există două domenii în care teoria negocierii se aplică expres pentru elaborarea deciziilor:

- elaborarea deciziilor multi-criteriale, domeniu ce implică negocierea deoarece există abordări diferite în privința criteriilor multiple precum ponderarea criteriilor, ierarhizarea criteriilor
- o *problemele concurențiale*, aceasta promovând scenariile în care anumite variabile ale deciziilor sunt controlate de de mai mulți factori independenți, fiecare dintre ei cu propriile lui interese.

Se presupune că toți jucătorii se comportă rațional, fapt credibil deoarece, după cum s-a arătat mai sus, când este vorba de interese de natură economică, oamenii se comportă rațional, chit că în situațiile de natură non-economică se comportă în mod emoțional. Comportamentul rațional înseamnă că fiecare individ își va alege astfel acțiunile încât să maximizeze utilitatea așteptată în ceea ce îl privește.

Negocierea este un proces care poate dura ceva mai multă vreme , ținând la elaborarea unui compromis între interesele părților implicate. În unele cazuri se face apel la o parte neutră, dar nu pentru a arbitra, ci pentru a ajuta ajungerea la un compromis prin sfaturile pe care le poate da în calitate de "cap limpede". Această persoană este, deci, un mediator, nicidecum un arbitru. Foarte adesea se face apel la un mediator, acesta fiind cazul negocierii cooperante, adică a negocierii asupra intereselor antagoniste în contextul voinței reale și puternice de a se aunge la înțelegere, părțile implicate punând înțelegerea mai presus de orice pentru promovarea intereselor lor

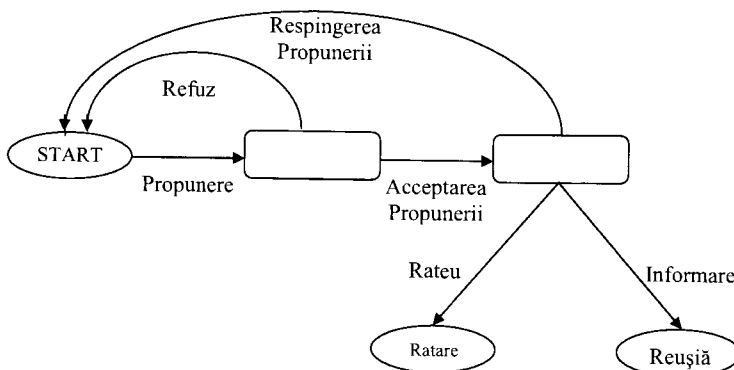
Sistemele multiagent trebuiesc configurate asemeni negocierilor dintre oameni, negocieri multi-laterale privinnd compromisul asupra mai multor obiective, aeasta cu atât mai mult cu cât funcția acestui sistem multi-agent este aceeași pe care programarea matematică o încearcă în cazul controlului mai multor factori de decizie răspândiți în teritoriu și care încearcă să gestioneze în mod optimal rețeaua de distribuție a energiei electrice.

2.4.Sistemul de negociere multiagent

Pe baza celor prezentate mai sus, se poate trece la abordarea chestiunilor privind implementarea sistemului multiagent, realizarea protocolului de negociere și atribuirea abilităților rațional-sociale sistemului multiagent.

2.4.1.Implementarea sistemului multiagent

Scopul este de a realiza o platformă de calcul ca sistem multiagent pentru controlul rețelei de distribuție a energiei electrice astfel încât să permită elaborarea deciziilor multi-criteriale de mai mulți factori de decizie răspândiți în teritoriul deservit de către compania distribuitoare de electricitate și urmărind fiecare mai multe obiective, atât comune, cât și diferite, platforma comunicând mesaje sincrone și asincrone, gestionând sarcini concurențiale cu mai multe protocoale de comunicare între cei care percep și emit informație celor care se află la distanță și cer acces la informație. Aplicatia trebuie să fie *orientata pe obiecte*, astfel încât să poată fi reprezentate diverscele entități ale rețelei de distribuție a energiei electrice precum furnizorii, producătorii, operatorii distribuției și alții.



- Fig.nr.2.1: Automatul pentru protocolul de realizare a contractului -

Aplicatia soft poate fi elaborata sub forma a opt pachete:

- clasele de bază ale agenților cooperanți,
- interfețele agenților,
- sarcinile executate de către agenți,

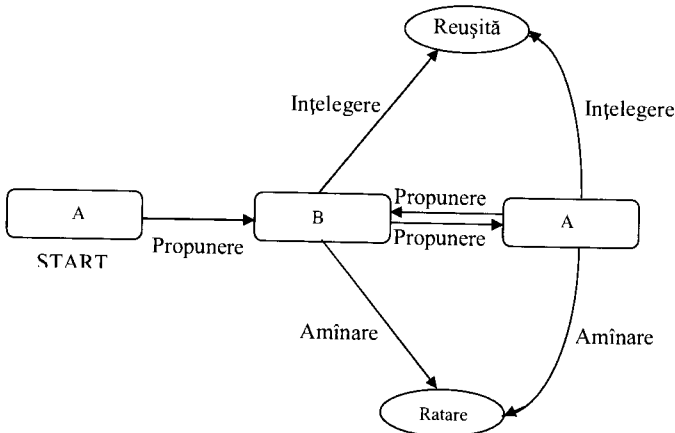
- o funcțiile de integrare agenți,
- o funcțiile pentru gestionarea protocoalelor,
- o interfețele serviciilor de coordonare,
- o interfețele pentru gestionarea mesajelor între agenți și
- o clasele abilitând negocierea între agenți.

Comunicarea între agenți este realizabilă prin folosirea unui mecanism inter-agent pentru mesaje, interpretarea mesajelor fiind specifică fiecărui agent, ceea ce îi oferă acestuia abilitatea de a interpreta același mesaj în funcție de starea internă a mașinii – vezi fig.nr.2.1. Structura mesajului, realizabilă conform recomandărilor FIPA-2000 [55], constă din expeditor, destinatar sau receptor, conținut, ontologie, identificator al conversației, protocol, replici și limbaj. Conversațiile multiagent sunt gestionate folosind "firele de execuție" cu ajutorul identificatorilor unici de conversație generați de către agentul care a inițiat conversația.

2.4.2. Protocolul de negociere

Se poate concepe o societate organizată de agenți pentru rețeaua de distribuție a energiei electrice, agenții în cauză fiind consumatorii, producătorii, transportorul și distribuitorul de electricitate, autoritatea reglementorie, vecinii distribuitorului cu care acesta face schimburi de electricitate. Deciziile trebuie să apară drept rezultat al negocierilor dintre agenți. Protocolul de negociere este bilateral, multi-obiectiv și integrativ, putând fi, deci, aplicat elaborării deciziilor atât în contextul incertitudinii, cât și în contextul certitudinii. Fie, deci, o negociere multilaterală, fără incertitudine, caz pe cât de general, pe atât de simplu. Evitând necesitatea modelului cu incertitudine, deciziile sunt elaborate pe baza ipotezelor privind *utilitatea* sau *valoarea* ori mai degrabă *utilitatea așteptată* sau *valoarea așteptată*. Atractivitatea acestei paradigme constă în garantarea încheierii cu succes a negocierii în virtutea *optimului Pareto* [53]. Procesul de negociere din acest caz, ilustrat în fig.nr.2.2, se desfășoară conform obișnuitelor tratative comerciale [59]. Agenții negociază valorilor unui set de obiective *independente preferențial* [44]. Când două obiective sunt independente preferențial, atunci fiecare agent își va exprima preferințele pentru (des)creșterea valorii unui obiectiv cu sau fără legătură cu celălalt obiectiv. Când sunt mai mult de două obiective, atunci independența preferențială este definită în mod similar pentru fiecare sub-set de obiective. Setul de obiective de negociat între două entități este stabilit pe baza câtorva noțiuni de genul *cantitate*, *calitate*, *preț unitar*, etc. Agenții negociază asupra valorii unui

obiectiv căruia i s-au precizat limitele, agenții înțelegându-se asupra limitelor valorii în cauză a fiecărui obiectiv în cursul unei pre/negocieri. Se presupune că sistemul multiagent S are cel puțin doi agenți, agentul a dorind să negocieze valoarea unui set de obiective cu ceilalți agenți, adică mulțimea $S - \{a\}$; fie deasemenea mulțimea obiectivelor $X_a = \{x_1, x_2, \dots, x_n\}$ pe care agentul a vrea să o negocieze, fiecare obiectiv din set avînd marja sa de valoare:

$$range(X_a) = \{[\min(x_1), \max(x_1)], \dots, [\min(x_n), \max(x_n)]\} \tag{2.1}$$


- Fig nr.2.2: protocolul de negociere al automatului cu stări finite -

2.5.Limbaje de comunicare inter-agent

Există mai multe limbaje de comunicare inter-agent, acestea fiind de fapt metalimbaje descriind comportamentul logic al agenților de către specialiști, ulterior textul fiind compilat în limbaj de programare de gen *Java* și apoi înserat în cod-mașină. S-a procedat astfel pentru a separa protocolul de comunicare dintre agenți, protocol ce trebuie să fie independent de domeniul de aplicare, de semantica mesajului de comunicat, mesaj al cărui conținut depinde de domeniul de aplicare [71].

2.5.1.Knowledge Query and Manipulation Language – KQML

La baza concepției acestui limbaj este analogia cu mai multe persoane vorbind limbi diferite. Când vorbște una dintre persoane, celelalte, chit că nu înțeleg nimic, își dau seama

că este vorba de comunicare în limbaj uman, un limbaj pe care l-ar putea învăța la nevoie. În schimb, pot memora mecanic mesajul și pot să-l transmită mai departe către persoanele care îi pot înțelege semnificația. De aici teza că, pentru a comunica, agenții trebuie să fie de acord asupra câtorva nivele de abstractizare:

- o transportul – cum emit sau recepționează mesajele,
- o limbajul – cum interpretează semnificația mesajelor,
- o politica – cum structurează conversațiile și
- o arhitectura – cum să fie arhitecturate sistemele în conformitate cu protocoalele de comunicare dintre agenți.

KQML este destinat schimbului de informație și cunoștințe, permițând astfel aplicațiilor să interacționeze cu sistemele inteligente. Limbajul a fost dezvoltat la început din inițiativa DARPA, devenind standard *de facto* pentru comunicația inter-agent [72].

2.5.2. Knowledge Interchange Format – KIF

Conținutul mesajului se referă în general la descrierea unor fapte care să fie înglobate în baza virtuală de cunoștințe a unui agent. Instrumentul matematic pentru reprezentarea cunoștințelor este logica simbolică. S-a observat că predicatul de ordinul întâi pot descrie cu suficientă acuratețe majoritatea informației de interes [71]. Caracteristicile esențiale pentru limbajul simbolic [73] și, deci pentru KIF, sunt următoarele:

- o semantică declarativă - sensul expresiilor poate fi înțeles fără ajutorul unui interpret,
- o comprehensiv – permite exprimarea propozițiilor logice arbitrare și
- o reprezentarea meta-cunoștințelor – utilizatorul poate, deci, descrie în mod explicit cunoștințele și poate introduce moduri noi de reprezentare ale acestora.

Mai pot fi introduse și alte caracteristici [73] precum:

- o translatabilitatea - posibilitatea traducerii bazei virtuale de cunoștințe în alte limbaje și alte moduri de reprezentare,
- o lizibilitatea – posibilitatea interacțiunii cu oamenii, chit că limbajul KIF nu a fost conceput pentru așa-ceva și
- o utilizabilitatea – posibilitatea de a fi folosit ca limbaj de reprezentare și / sau comunicare în cadrul aplicațiilor, chit că limbajul KIF nu a fost conceput pentru așa-ceva.

KIF a fost propus ca standard pentru descrierea cunoștințelor din sistemele expert, bazele de date, agenții inteligenți, etc. Acest limbaj este versiune a calculului predicativ de ordinul întâi,

cu extensii ce suportă raționamente nemonotone și definiții. De pildă, un automobil poate fi descris prin tripleta număr, culoare și marcă.

2.5.3. Coordination Language – COOL

În mediile cu resurse limitate, agenții trebuie să-și coordoneze activitățile pentru promovarea propriilor interese și îndeplinirea scopurilor de grup. Acțiunile agenților trebuie coordonate deoarece acestea sunt interdependente și nici un agent nu are competența, resursele și cunoștințele necesare atingerii de unul singur a țintelor sistemului. Dacă KQML se adresează nivelului intențional al interacțiunii dintre agenți, iar KIF se referă la conținutul informațional al mesajului, atunci COOL tratează coordonarea și de aceea este utilizat pentru proiectarea, reprezentarea și validarea mecanismelor și protocoalelor de coordonare în sistemele multiagent. Activitatea de coordonare este modelată ca o *conversație* între doi sau mai mulți agenți [56], conversație specificată cu logica automatului cu stări finite:

- stările automatului reprezintă stările conversației; există o stare inițială distinctă în care începe conversația și câteva alte stări care semnalează încheierea conversației
- mesajele schimbate sunt reprezentate de informația privind structura mesajului
- mulțimea regulilor de conversație specifică modul în care un agent aflat într-o anumită stare primește mesaje de un anumit tip, execută acțiuni locale, emite mesaje și trece într-o altă stare
- mulțimea regulilor de refacere în caz de eroare arată tratamentul incompatibilității dintre starea curentă a conversației și mesajele primite
- mulțimea regulilor de continuare arată modul de acceptare de către agenți a cererilor de conversație și modul de selectare a continuării conversației
- clasele conversației includ stările, regulile conversației și regulile erorilor specifice unui tip de conversație
- conversațiile efective instațiază clasele de conversație și sunt create oricând, agenții angajându-se într-un proces de comunicare.

Informații privind structura mesajului KQML i-au fost adăugate câteva elemente noi:

- *propose* – propune atingerea unei ținte – vezi fig.nr.2.1 și 2.2
- *counter-propose* – contra-propunere ca răspuns la propunerea inițială ca urmare a faptului că destinatarul nu poate atinge ținta propusă
- *accept, reject* – acceptarea / respingerea (contra-)propunerii
- *satisfy, fail* – succesul / eșecul acțiunii.

2.6. Sisteme blackboard

Comunicarea între agenții sistemului multiagent se poate face în mai multe feluri, fie prin mesaj către un anumit destinatar, fie prin mesaj către toți agenții, *broadcast*, răspunsul fiind exprimat în modul ilustrat mai sus. Există și varianta, deasemenea ilustrată în amintitul subcapitol, comunicării mesajului unui agent special, acel gestionar central de siguranță, care se va ocupa cu rolul de "poștaș". În sfârșit, există și posibilitatea unei baze de date comune [71] în care agenții pot scrie și intercomunica conform descrierii următoare:

"Să ne imaginăm un grup de specialiști așezați lângă o tablă mare. Ei lucrează împreună pentru rezolvarea unei probleme, scriind pe tablă pentru a dezvolta soluția. Mai întâi sunt incipiente problema și datele inițiale. Specialiștii urmăresc tabla, căutând o ocazie pentru a-și pune în valoare cunoștințele. Când un specialist capătă suficientă informație pentru a-și aduce aportul, el își înregistrează contribuția pe tablă. Această informație suplimentară va permite altor specialiști să-și aplice cunoștințele. Procesul adăugării pe tablă va continua până ce problema va fi rezolvată."

Acesta este principiul sistemelor *blackboard*, cuvânt semnificând "tabla școlară".

Odată definit principiul sistemului *blackboard* în interesul intercomunicării performante între agenții sistemului multiagent, caracteristicile se impun de la sine:

- independența cunoștințelor – mai-sus amintiții specialiști, denumiți și "surse de cunoaștere", nu sunt instruiți să lucreze exclusiv într-un anumit grup, fiecare fiind expert într-un sector al problemei și putând contribui la soluție independent de ceilalți,
- diversitatea tehnicilor de rezolvare – reprezentarea internă și mecanismelor de inferență utilizate de către sursele de cunoștințe sunt diferite de la o sursă de cunoaștere la alta, de la un agent la altul,
- limbajul comun de interacțiune – agenții trebuie să interpreteze corect informația, în realitate fiind vorba de un compromis între expresivitatea reprezentării specializate, accesibilă unui număr redus de agenți, și expresivitatea reprezentării generale, înțelegează de către toți agenții,
- activitatea pe baza evenimentelor – agenții sunt activați de evenimentele ce apar scrise pe "tablă", etc.,
- necesitatea controlului se impune prin imperativul unui responsabil al administrării procesului de rezolvare a problemei și

- generarea incrementală a soluției – procesul de rafinare a soluției, contrazicând informația existentă și inițiind o nouă cale de raționament.

Capitolul 3

Sistem multiagent cu autonomie completa pentru restabilirea alimentarii în rețelele de distribuție a energiei electrice

3.1. Problema restabilirii rețelei de distribuție a electricității în urma unei avarii

Energia electrică prezintă o serie de avantaje în comparație cu alte forme de energie, dar și o bună siguranță în funcționare. Prin siguranța în funcționare se definește aptitudinea unui dispozitiv sau a unei instalații de a-și îndeplini funcția specificată în condițiile date. Principalele aspecte care caracterizează siguranța în funcționare, respectiv continuitatea în alimentarea cu energie electrică a unui consumator sunt:

- numărul anual de întreruperi determinate de reparații, respectiv de manevre în instalații;
- durata medie a unei întreruperi;
- durata de restabilire a alimentării cu energie electrică;
- durata totală medie de întrerupere pe an.

Dintre toate acestea, cel mai important aspect îl reprezintă durata de restabilire a alimentării cu energie electrică, de cele mai multe ori fiind considerat un indicator de performanță atât pentru cei ce exploatează rețelele de distribuție a energiei electrice, cât și pentru autoritățile de reglementare.

Restabilirea după o avarie înseamnă reenergizarea liniilor și barelor care au fost decuplate de la energia electrică de către sistemele de protecție, în urma apariției unei avarii. Aceasta înseamnă izolarea unui defect și restabilirea rețelei de distribuție a energiei electrice astfel încât să se asigure continuitatea în alimentarea cu electricitate a consumatorilor. Sistemul de restabilire trebuie să ia un număr de decizii și să le execute, astfel încât zonele neenergizate să fie alimentate de la cele învecinate care pot să asigure necesarul de putere. Eventual anumiți consumatori pot fi decuplați.

În momentul în care rețeaua de distribuție a electricității este întreruptă de apariția unui defect, devine necesară restabilirea optimă în raport cu defectul apărut. Problema restabilirii rețelei de distribuție a electricității implică diferite abordări, acestea putând fi clasificate în patru categorii: metode euristice, sisteme expert, programare matematică și "soft computing". Metodele euristice și sistemele expert au fost des utilizate în industrie, dar sunt deficiente în

ceea ce privește caracterul optimal al soluțiilor. Programarea matematică este capabilă să obțină soluția optimă, dar necesită de obicei un timp de execuție prea mare față de restricțiile de timp, atât de natură comercială, cât și de natură tehnică. Restricțiile comerciale de timp sunt impuse de reglementările care limitează durata maximă de întrerupere, numărul de întreruperi și durata totală anuală de întrerupere în alimentarea cu electricitate a consumatorului. Imperativul restricțiilor tehnice de timp este acela de a evita premisele "avariei în avalanșă", adică acea situație în care un defect eminent local poate provoca prăbușirea întregii rețele—vezi celebra avarie din România din 10 mai 1977 și avariile din SUA din ultima vreme. Deși metoda "soft computing" este ușor de implementat, e deficentă pe de o parte prin aceea că nu se poate obține pe baza acesteia soluția optimă în adevăratul sens al cuvântului și mai ales pentru că durata de elaborare a deciziei este prea mare față de restricțiile de timp impuse uzual. Așadar, se impunea o altă soluție, aceea a inteligenței artificiale.

3.2. Modelul matematic privind restabilirea rețelei de distribuție a electricității

Obiectivul modelului matematic privind restabilirea rețelei de distribuție a electricității după o avarie este maximizarea puterii absorbite din rețeaua de distribuție a electricității. Evident, valoarea este puterea maxim-absorbită de către toți consumatorii la un moment dat. Funcția obiectiv va fi:

$$\max \sum_{k \in R} L_k \cdot y_k \quad (3.1)$$

unde L_k este sarcina de pe magistrala k , y_k este variabila de decizie, $y_k=1$ - linia electrică re-energizată, $y_k=0$ - linia electrică ne-energizată, și R reprezintă mulțimea sarcinilor ne-energizate. Restricțiile tipice care caracterizează modelul de restabilire a rețelei de distribuție a electricității sunt următoarele:

(a) limitarea ca și capacitate a surselor de putere disponibile pentru restabilirea rețelei:

$$\sum_{e \in F_q} P_e \cdot x_e \leq G_q \quad (q \in S) \quad (3.2)$$

unde P_e este fluxul de putere în direcția liniei electrice e , presupunând că $P_e \geq 0$, x_e variabila de decizie corespunzătoare lini electrice e , $x_e=1$ - linia electrică e este inclusă în calea de re-energizare, $x_e=0$ - linia electrică e este exclusă de pe calea de re-energizare, F_q mulțimea

de linii conectate la bara q, G_q puterea din bara energizată q și S setul de linii electrice energizate.

(b) balanța energetică între puterea cerută și puterea disponibilă pe liniaelectrică i este:

$$\sum_{k \in T_i} P_k - \sum_{k \in F_i} P_k - L_i \cdot y_i = 0 \quad (i \in N) \quad (3.3)$$

unde T_i este puterea disponibilă pe linia electrică i , F_i este puterea cerută pe linia electrică i și N numărul de linii electrice energizate.

(c) limitarea puterii pe liniile electrice:

$$|P_k| - U_k \leq 0 \quad (k \in B) \quad (3.4)$$

unde P_k este puterea vehiculată pe linia electrică k , U_k puterea nominală a liniei electrice k și B mulțimea liniilor electrice.

(a) restricțiile în configurație radială se referă la faptul că trebuie obținută o configurație radială, cerință obligatorie în operarea rețelelor actuale de distribuție a electricității. Pentru a asigura o configurație radială, numărul total de linii electrice i conectate la o bară trebuie să fie cel mult 1, adică:

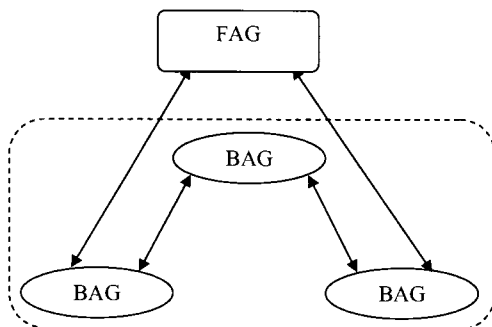
$$\sum_{k \in T_i} x_k \leq 1 \quad (i \in N) \quad (3.5)$$

3.3. Arhitectura sistemului multiagent cu autonomie completa

De fapt, agenții inteligenți sunt utilizați într-o mare varietate de aplicații. Multe aplicații importante din domeniul informaticii, ca de exemplu planificarea, conducerea proceselor, configurarea rețelelor de comunicații și sistemele concurente pot beneficia de pe urma unei abordări de tip sistem multiagent [98], [81], [78], [94]. Un sistem multiagent este un sistem computațional în care mai mulți agenți cooperează pentru îndeplinirea unor sarcini. El are la bază caracteristici precum interacțiunea, colaborarea și autonomia. Sistemul multiagent folosit la lichidarea avariilor trebuie să se caracterizeze printr-un număr redus de tipuri de agent, tocmai pentru a i se garanta eficiența, adică elaborarea deciziilor într-un interval de timp mai mic decât restricțiile tehnice și/sau comerciale.

În lucrarea [94] se propune o arhitectura de sistem multi-agent pentru rezolvarea problemei restabilirii alimentării după o avarie. În cadrul prezentei lucrări, pornind de la aceasta arhitectura, se va dezvolta în capitolul 4 un sistem multiagent ce constituie o implementare a acestei arhitecturi.³

Se propune o arhitectura (vezi fig. 3.3) constând dintr-un număr de agenți de magistrală (Bus AGent –BAG), pe care îi vom numi în această lucrare *agenți-bară* și un sigur *agent facilitator* (Facilitator AGent –FAG), numit în această lucrare *agent-dispecer*. Practic, fiecarei bare de pe același nivel de tensiune a unui sub-sistem de distribuție a electricității i se asociază un *agent-bară* ca agent inteligent. *Agentul-dispecer* este asociat dispecerului ce are autoritate de decizie asupra subsistemului de distribuție a electricității. O astfel de arhitectura am numit-o în această lucrare cu *autonomie completa*, pentru ca în afara de agentul facilitator, ceilalți agenți sunt omogeni și nu au nicio ierarhie între ei, funcționând ca entități autonome din punctul de vedere al deciziilor luate.



- Figura nr.3.3: arhitectura sistemului

Implementarea arhitecturii multiagent pentru restabilirea rețelei de distribuție a electricității utilizează tehnica de proiectare orientată pe obiect.

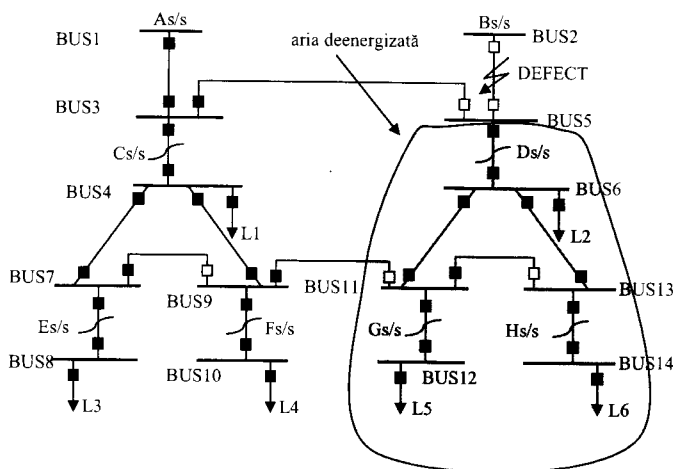
Rolul unui agent BAG este de a căuta o sursă pentru realimentarea consumatorilor direct conectați la bara careia îi este asociat. Agentul BAG are la bază un set de reguli simple pentru reconectarea consumatorilor.

3.4. Procesul de negociere

Procesul de negociere este unul dintre procesele cheie pentru sistemul multiagent în atingerea scopului propus. Pentru a ilustra acest proces vom utiliza modelul de rețea din fig.3.4.cu 8 stații (A s/s – H s/s) și 14 magistrale (BUS). Patratelele reprezintă întrerupătoare care au starea închis, dacă sunt negre, sau deschis, dacă sunt albe. Eticheta "BUSxx" aparține barei cu numărul "nn", iar etichetele "Ln" aparțin consumatorilor magistralei n.

Subsistemul de-energizat prin avarie este marcat cu aria gri din fig.3.4. Linia dintre stațiile

Bs s și Ds s este întreruptă din cauza defectului presupus și trei consumatori, $R=\{L2, L5, L6\}$, trebuie să fie realimentați de sistemul multiagent. În acest caz particular BUS3 și BUS9 au putere disponibilă pentru restabilire ceea ce se notează cu $S=\{BUS3, BUS9\}$.



- Figura nr.3.4: procesul de restabilire a rețelei de distribuție a electricității -

Secvența de negociere este ilustrată în fig.nr.3.5. În această figură, dreptunghiurile hașurate reprezintă agenții BAG corespunzători barelor neenergizate, iar celelalte dreptunghiuri reprezintă agenții BAG asociați barelor energizate.

Capitolul 4

Implementarea unui sistem multiagent cu autonomie completa pentru restabilirea alimentarii utilizand mediul de dezvoltare JACK

Scopul acestui capitol este implementarea sistemului multiagent descris în capitolul anterior, care să realizeze procesul de negociere automată și restabilirea subsistemului de distribuție a electricității așa cum a fost descris în acest capitol. Pentru aceasta s-a folosit sistemul de dezvoltare multiagent JACK deoarece este o platforma prietenoasă și bine structurată teoretic.

4.1. Implementarea sistemelor multiagent utilizand platforma JACK

Paradigma orientării-agent a apărut din necesitatea unei inteligențe artificiale distribuite, cu putere de raționament și decizie în situațiile proiectate. JACK Intelligent Agents de la compania Agent Software este un mediu vizual de programare cross-platform construit pe baza limbajului *Java*. El conține toate componentele *Java*, adăugând în plus extensii specifice pentru lucrul cu agenți. JACK a fost creat pentru a aduce limbajului *Java* extensia orientării agent. Noțiunea de agent presupune entități autonome capabile să ia decizii prin reacție la evenimentele generate din mediu. Este foarte utilă în multe zone de aplicabilitate, precum sistemele business distribuite, control și comanda, aplicații inteligente și simulări. Deși aflat la începuturile sale, sistemul s-a dovedit fiabil în rezolvarea problemelor de organizare a traficului aerian. Un **agent** încapsulează comportamentul dorit în unități ce conțin toate definițiile și structurile necesare pentru a funcționa independent. Agenții se situează la un nivel superior conceptului de programare orientată-obiect. Aceasta componenta de baza a mediului JACK manifestă putere de raționament proprie, putând funcționa atât în modul de a urmări un scop predefinit, cât și reacționând la evenimente apărute spontan. Fiecare agent are în componența sa:

- un set de **credințe** despre lume (de exemplu setul sau de date)
- un set de **evenimente** la care va răspunde
- un set de **scopuri** pe care dorește să le atingă, fie conform credințelor sale fie ca urmare a evenimentelor apărute
- un set de **planuri** care descriu modul în care agentul rezolvă problema apărută

Când un agent este instanțiat în sistem, el așteaptă apariția evenimentului pentru care a fost creat și în acel moment determină soluția problemei analizând planurile sale pentru a-l găsi pe cel relevant și aplicabil situației. În cazul în care nu-i reușește planul inițial, el caută alte alternative de rezolvare până când reușește, sau toate variantele se epuizează. În cadrul unui singur proces pot exista mai mulți agenți, fiecare având potențial multiple cozi de sarcini. O astfel de coadă se generează când apare un eveniment asincron și conține pașii necesari pentru rezolvarea sa. Dacă evenimentul este postat sincron, sarcinile rezultate sunt plasate în vârful cozii care a generat evenimentul. Metoda de programare orientată agent devine și mai interesantă când sunt creați agenți interconectați în diferite procese și pe diferite mașini.

4.2. Proiectarea și implementarea sistemului multiagent cu autonomie completă

Proiectarea aplicației multiagent s-a făcut ca o primă încercare de a realiza un sistem multiagent dedicat problemei restabilirii structurii unui subsistem de DEE. Contextul este cel prezentat în capitolul precedent, iar implementarea este tributară sistemului de dezvoltare JACK, așa cum a fost prezentată în subcapitolul precedent.

4.2.1. Definierea agenților

Din capitolul 3 rezulta că se va considera în sistem un număr de $n+1$ agenți, unde n este numărul de magistrale (bare de tensiune) din sistem

- un agent facilitator FAG;
- n agenți asociați magistralelor avariate sau nu, numite BAGi

Aplicația urmărește să realizeze implementarea procedurilor de negociere și de reacție la evenimente ce caracterizează procesul de restabilire. Se presupun cunoscute următoarele date:

- lista *DEBList* a magistralelor implicate în avarie;
- lista *ListaAvarie* care este un duplicat al listei *DEBList* și folosește la determinarea momentului când BAGs și-au determinat puterile cerute pentru restabilire și pot începe negocierile;
- lista *PosibW* pentru fiecare BAG, lista ce conține identitate BAG-urilor asociate magistralelor neavariate care pot energiza magistrala în cauză. Acestea se determină într-o etapă anterioară începerii negocierilor

- lista *ListaVecini* pentru fiecare BAG, lista ce conține identitate BAG-urilor asociate magistralelor avariate care pot primi putere de la BAG-ul respectiv.

4.2.2. Definirea evenimentelor și a mesajelor asociate acestora

Evenimentele definite prin implementare în sistemul multiagent sunt evenimente de tip eveniment-mesaj. Acestea sunt:

- **SIMULARE**: eveniment primit de FAG pentru a porni restabilirea
- **AVARIE**: trimis de FAG către toate BAG-urile pentru a se pregăti de restabilire
- **COALITIE**: trimis de un BAG ce se poate energiza doar de la o magistrală și această avariata către BAG-ul asociat acesteia, în vederea stabilirii unei coaliții în care să fie reprezentat de către acesta din urmă.
- **GATA**: trimis de fiecare BAG către FAG pentru a marca faptul că este pregătit pentru negocieri
- **ACK**: trimis de BAG-ul ce acceptă coaliția către BAG-ul ce a cerut-o.
- **OK**: trimis de fiecare BAG către FAG când este energizat (restabilit)
- **START**: trimis de FAG către BAG-ul ales să înceapă restabilirea prin negocieri.
- **NEG-IA**: trimis de un BAG avariata către un BAG ce poate da energie din afara zonei avariate (*PosibW*).
- **CONF-IA**: trimis de BAG-ul ce poate da energie din afara zonei avariate către BAG-ul avariata n semn că i se confirmă cererea de putere.
- **START-NEG-DA**: trimis de un BAG avariata restabilit către el însuși pentru a inspecta *ListaVecini*
- **NEG-DA**: trimis de un BAG avariata restabilit către un BAG vecin avariata (din *ListaVecini*) cu o oferta de putere disponibilă.
- **CONF-DA**: trimis de un BAG avariata restabilit către un BAG restabilit ce i-a oferit putere, cu precizarea puterii luate.

Mesajele asociate evenimentelor conține anumite informații, așa cum sunt prezentate mai jos:

COALITIE—identitatea expeditorului

GATA—identitatea expeditorului

ACK—identitatea expeditorului

OK—identitatea expeditorului

START—identitatea expeditorului și puterea cerută

NEG-IA—identitatea expeditorului și puterea ceruta

CONF-IA—identitatea expeditorului și puterea cedată

NEG-DA—identitatea expeditorului și puterea disponibilă

CONF-DA—identitatea expeditorului și puterea necesară.

4.2.3. Definirea planurilor de tratare a evenimentelor

Fiecărui eveniment îi asociem în acest sistem câte un singur plan de tratare. Acestea sunt descrise în cele ce urmează:

Plan tratare eveniment SIMULARE

FAG trimite eveniment-mesaj **AVARIE** la toate BAG-urile implicate în avarie. Acestea se găsesc în lista *DEBList*.

Plan tratare eveniment AVARIE

determinarea puterii ceruta P_{ceruta} de către agentul în cauză pentru restabilire;

dacă există o singură bară externă care îi poate energiza propria bară,

atunci trimite eveniment-mesaj **AVARIE** însoțit de P_{ceruta} către BAG al barei externe,

altfel trimite eveniment-mesaj **GATA** către FAG.

Plan tratare eveniment COALITIE

se corectează $P_{ceruta} = P_{ceruta} + P_{ceruta-BAG}$;

se trimite eveniment-mesaj **ACK** către BAG în cauză, în semn de acceptare a coaliției, și

se marchează în datele interne coaliția;

Plan tratare eveniment ACK

marchează faptul că este în coaliție, înscrind asta în datele sale interne;

trimite mesajul **GATA** către FAG.

Plan tratare eveniment GATA

mesajul este primit de către FAG care:

identifică BAG expeditor și-l șterge din ListaAvarie

dacă ListaAvarie = \emptyset

atunci trimite eveniment-mesaj **START** către BAG în cauză

DEBList este ordonată corespunzător și practic se ia primul din listă.

Plan tratare eveniment **START**

selectează din lista *PosibW* primul BAG și

trimite la acest BAG evenimentul-mesaj **NEG-IA** cu mesaj ce conține puterea cerută.

Plan tratare eveniment **NEG-IA**

dacă $P_{\text{disponibila}} > P_{\text{ceruta}}$,

atunci trimite evenimentul-mesaj **CONF-IA** la BAG încauză,

altfel trimite evenimentul-mesaj **NU-POT** la BAG în cauză.

Plan tratare eveniment **CONF-IA**

marchează faptul că bara a fost energizată și calculează

$P_{\text{disponibila}} = P_{\text{disponibila}} - \text{sarcina}$,

apoi trimite evenimentul-mesaj **START-NEG-DA**

Plan tratare eveniment **START-NEG-DA**

preia primul BAG din lista *ListaVecini* și îi trimite un evenimentul-mesaj **NEG-DA** ce conține puterea disponibilă.

Plan tratare eveniment **NEG-DA**

dacă $P_{\text{necesara}} \leq P_{\text{disponibila(oferita)}}$

Atunci

{

se automarchează ca energizat

trimite evenimentul-mesaj **CONF-DA** către BAG
ofertant
trimite evenimentul-mesaj **OK** la FAG.
}

Plan tratare eveniment **CONF-DA**

recalculează $P_{\text{disponibila(oferita)}} = P_{\text{disponibila(oferita)}} - P_{\text{neccesara}}$
șterge BAG expeditor din lista *ListaVecini*
trimite evenimentul-mesaj **START-NEG-DA** către el
însuși.

4.3 Concluzie

Acest capitol a permis studiul și utilizarea platformei JACK pentru realizarea unui sistem multiagent de restabilire după avarie a RDEE. Comportamentul propus în capitolul 3 a fost repercutat în organizarea agenților și dotarea cu inteligență a acestora. Contribuțiile propuse s-au materializat în

- definirea agenților,
- definirea evenimentelor și a mesajelor asociate acestora,
- definirea planurilor de tratare a evenimentelor.

Aceste aspecte implementează ceea ce de manieră generală numim comunicare și negociere între agenți. Aspectele legate de credințe (convingeri) sunt relativ simple, în acest caz, și se materializează prin datele pe care le încapsulează agenții.

Capitolul 5

Sistem multiagent cu autonomie supervizată pentru restabilirea alimentării cu energie electrică

Daca în capitolele 3 și 4 a fost prezentat și realizat un sistem multiagent în care autonomia agenților în negociere era practic *completa*, în acest capitol se propune o soluție mai realistă, în care agenții utilizează și experiența de restabilire acumulată în dispecerate și în care autonomia este *supervizată*. Ierarhia este folosită pentru a reduce timpul de cautare prin negociere și pentru ca sistemul să se îndrepte rapid către o soluție fezabilă. Rezultatul propunerii este Sistemul MultiAgent pentru Restabilirea Alimentării cu energie electrică - SMARA. Comunicarea și negocierea între agenți au, în timp, un caracter schimbător, în sensul că, în timp, SMARA trece de la o structură fără ierarhie la una ierarhizată a agenților. Această trecere se face datorită faptului că sistemul de restabilire adoptă potențial unul din cele trei moduri de operare: *auto-operare*, *operare ierarhizată* și *operare arbitrată*, descrise în secțiunea următoare. În acest capitol, prin nod se înțelege agentul magistrală sau bara asociat cu bara de tensiune a unei stații a RDEE.

5.1. Modurile de operare ale nodurilor rețelei de distribuție a energiei electrice

SMARA este organizată ca suport pentru operarea inteligentă a lichidării avariilor în rețelele de distribuție a energiei electrice și, deci, de restabilire a alimentării cu energie electrică a consumatorilor. Pentru aceasta se consideră că topologia rețelei de distribuție este compusă din elemente de rețea:

- autonome=nodurile rețelei—clasă de obiecte ale căror proprietăți asigură caracteristici BDI, comportamentul autonom fiind diferențiat în funcție de situația energetică a nodului:
 - noduri consumatoare ≡ barele S.T.110 kV / M.T. și / sau ale P.T.-M.T./ J.T., al căror comportament BDI este definit prin:
 - noduri generatoare ≡ numai pentru generatoarele distribuite în rețeaua de distribuție a electricității, al căror comportament BDI este definit prin: și
 - "dispecerul" ≡ obiect virtual arbitrând jocul autonom al nodurilor rețelei și al cărei comportament BDI este definit prin:
- elemente de rețea non-autonome ≡ clasă de obiecte cu proprietăți diferite, toate lipsite de

comportament BDI și, implicit, de comportament autonom și care se supun, deci, "intențiilor" elementelor autonome:

- linii electrice aeriene și / sau subterane,
- aparate de comutație primară – întrerupătoare, separatoare de sarcină, separatoare și re-anclanșatoare,
- baterii de condensatoare,
- transformatoare de putere,
- alte elemente ale rețelei de distribuție a electricității

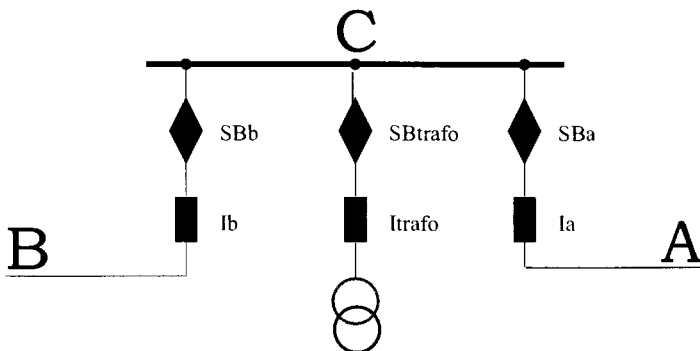
Nu se acceptă "restricții", ci teza unei "restricții unice și complexe" integrând restricțiile fizice printr-o schemă logică de tipul "if . . . then . . . elseif . . . then . . . else", indiferent cât de complexă ar fi această schemă, aceasta definind de fapt "misiunea" agentului inteligent într-o situație complexă în cursul căreia informația este livrată în avalanșă. Practic, această "restricție unică" revizuieste "convingerile" în funcție de starea conjuncturală a rețelei printr-un proces iterativ, generează opțiuni pentru dorințe, acestea fiind filtrate pentru a defini **intenția** agentului inteligent printr-un proces de asemenea iterativ.

În această ipoteză, operarea inteligentă va avea loc pe trei nivele, în funcție de "intenții"

5.1.1 Auto-operarea

În acest caz nodul consumator sau generator având un vecin energizat, se adresează vecinilor săi în vederea accesului la tensiune.

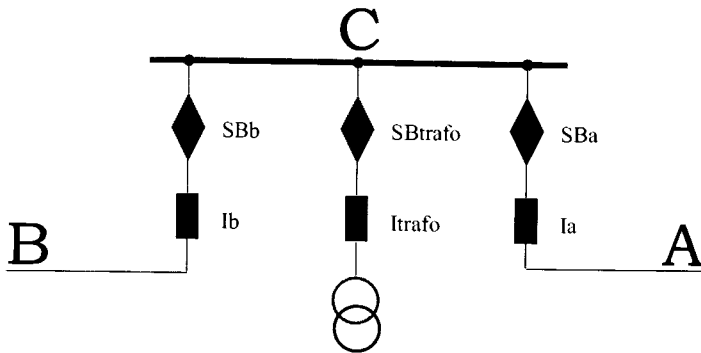
- nodul consumator a rămas fără tensiune după cum se poate observa în figura nr.5.1.



- Fig.5.1: nodul consumator a rămas fără tensiune -

Nodul consumator **C** e alimentat din nodul vecin **A** și la un moment dat este dezergizat. Asemeni dispozitivelor AAR, nodul consumator **C** va deconecta întrerupătorul dinspre vecinul **A** și va conecta întrerupătorul dinspre vecinul **B**. Are loc energizarea nodului consumator **C**.

Rezultatul este prezentat în figura nr.5.2, caz în care "convingerea" nodului **C** este aceea că trebuie să găsească tensiune pentru consumatorii săi, "dorința" fiind aceea că va găsi tensiune în nodul **B**,



- Fig.nr.5.2: Realimentarea cu energie electrică a nodului consumator - iar "intențiile", în funcție de situația conjuncturală, vor fi:

- realimentarea consumatorilor în caz de succes
sau
- declinarea competenței în caz de eșec

5.1.2 Operarea ierarhizată

Implică intervenția "dispecerului" la **cerere**. Atunci când mai multe noduri / bare îl anunță că nu au găsit tensiune, motiv pentru care și-au declinat competența, caz în care "restricția unică" după care se ghidează este următoarea:

"Convingerea" elementului autonom "dispecer" este aceea că, procesînd succesiv nodurile ce și-au declinat competența, există șansa ca unul dintre acestea să fi găsit un vecin energizat în cursul unei secvențe de auto-operare și, confirmate între timp de către utilizator ≡ dispecerul uman, poate re-energiza toate nodurile.

"Dorința" elementului autonom "dispecer" este aceea de energiza în avalanșă nodurile ce

și-au declinat competența, alegînd și învățînd ordinea oportună.

"Intenția" agentului autonom "dispecer" este aceea de a energiza nodurile din amonte în aval, pornind de la sursa sub-rețelei de distribuție a electricității către cel mai apropiat nod și apoi, din aproape în aproape, către cel mai îndepărtat nod.

5.1.3 Operarea arbitrată

Implică participarea activă a "dispecerului" ca agent autonom, **nu la cererea** nodurilor ce și-a declinat competența, ci din **proprie inițiativă** ca urmare a faptului că apar zone topologice izolate prin "condamnarea" unora dintre aparatele de comutație primară din cauza apariției unor defecte ireversibile și, deci, fără acces la tensiune, ceea ce implică intervenția la nivelul de tensiune superior—vezi figura nr.5.6. "Dispecerul" va desfășura după cum urmează:

- identifică și izolează sub-rețeaua calamitată procesînd exclusiv topologia rămasă
- activează logic toate sursele de tensiune ale sub-rețelei de distribuție în cazul în care dispecerul are autoritate de decizie și la nivelul de tensiune imediat superior
- elaborează coada de așteptare virtuală pentru fiecare sursă de tensiune reală și logică
- procesează aceste cozi de așteptare, mai întîi pentru sursele reale, apoi pentru cele logice, în mod similar operării ierarhizate, pînă ce a energizat toate nodurile din afara zonei calamitate, (in)validînd nodurile logice; în același timp sunt eliminate din cozile de așteptare acele noduri care au fost energizate deja prin procesarea precedentelor cozi de așteptare
- energizează nodurile logice la nivelul de tensiune superior
- predă controlul interfeței om-mașină, procesînd mai întîi la nivelul superior de tensiune, când e cazul, și apoi la cel inferior

În cazul acestui mod de operare, "convîngerea" agentului autonom "dispecer" este aceea toate nodurile sub-rețelei din afara zonei calamitate pot fi energizate atunci când sunt energizate sursele de tensiune ale acesteia, "dorința" agentului autonom "dispecer" fiind, în consecință, aceea de a elabora în cozile de așteptare cu maximum de noduri posibile, astfel încît să nu fie nevoie de a energiza toate nodurile logice-surse de tensiune pentru sub-rețeaua aflată în dificultate; de aici și "intenția" **agentului autonom** "dispecer" de a energiza maximum posibil de noduri în sub-rețeaua aflată în dificultate, energizînd minimum de noduri-sursă de tensiune ale acestei sub-rețele de distribuție a electricității.

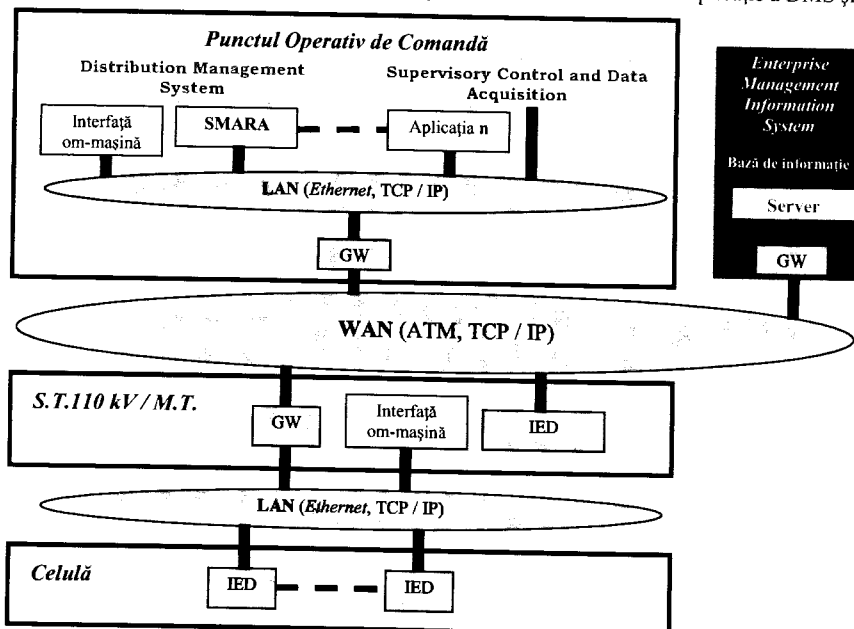
SMARA evită orice calculație tocmai pentru a răspunde prompt în timp real deoarece, spre deosebire de aplicațiile DMS **asistînd** dispecerul, SMARA se desfășoară sub presiunea

imperativului realimentării consumatorului. Gestionarea cozii de așteptare este esențială, în care scop se folosesc următoarele mecanisme:

- alterarea priorității nodului în cursul operării în funcție de (ne)reușita energizării sale,
- agregarea restricțiilor într-o restricție unică de genul *if...then...*, inclusiv aceea de eliminare din topologia operativă a zonei calamitate din rețeaua de distribuție a electricității,
- constituirea cozii de așteptare prin metode euristice de genul "sacului de monezi false",
- procesarea ciclurilor imbricate și cu multe iterații cu noile tehnici de găsim a valorii căutate, tehnici pe baza segmentării vectoriale \equiv similar căutării defectului pe cablu,
- schimbul de informație între cozile multiple prin tehnici de genul *send by receive*

5.2. Organizarea aplicației SMARA

Locul SMARA în arhitectura sistemului autonom de monitorizare și control a distribuției energiei electrice este ilustrat în figura nr.5.11. SMARA este o aplicație a DMS și



- Fig.5.11: Locul SMARA în sistemul autonom de monitorizare și control -

e folosită decât de utilizator, dispecerul avînd autoritatea operativă de decizie în rețeaua de distribuție a energiei electrice, pentru restabilirea distribuției energiei electrice—RDEE atunci când e cazul. SMARA este conectată, prin intermediul DMS, la baza de informație, astfel nîcît "cunoaște" în orice moment starea reală a topologiei rețelei de distribuție a energiei electrice. DMS are mai multe funcții, acestea constituind aplicațiile DMS și fiind realizate fiecare cu cîte un modul al aplicației:

- ❑ *state processor*—modul procesînd starea topologiei rețelei în funcție de datele din baza de informație, date actualizate în timp real, unele *on-line* altele *off-line* prin intermediul interfațe om-mașină
- ❑ *power flow*—circulațiile din rețeaua de distribuție a energiei electrice, vizualizate într-o schemă cinematică pe ecranul stației de lucru sau al unui *video-wall*; modulul interacționează și cu SMARA (in)validînd o manevră propusă pe considerente de supra'cîrcare a unei linii electrice, tocmai pentru a garanta continuitatea alimentării consumatorilor.
- ❑ *short-circuits analysis*—analiza scurt-circuitelor, adică a defectelor ireversibile, modul care, între altele, "condamnă" anumite aparate de comutație primară în vederea îuolării defectelor zona,avînd drept rezultat în baza de informație definirea "zonei calamitate" pentru ca SMARA să funcționeze corect
- ❑ *voltage management*—stabilește nivelul tensiunilor pe fiecare element de rețea pe baza rezultatelor circulațiilor puterilor, verificînd încadrarea tensiunii în bandă și sugerînd măsurile necesare, modulul interacționînd cu SMARA (in)validînd manevrele propuse.
- ❑ *network recovery*—este funcția prin care se realizează RDEE și nu dă suficientă satisfacție la funcționarea în timp real deoarece are întîrziiri în ciuda resurselor consumate, explicația fiind calculația mult prea complexă.

Așadar problema de rezolvat era elaborarea și implementarea modulului *network recovery* al DMS sub forma SMARA, ceea ce implica:

- ❑ conceperea unui proces autonom de analiză a topologiei rețelei de distribuție a energiei electrice, proces lansat în momentul dezergenizării unui nod și care să identifice soluțiile de energizare a nodurilor aflate în dificultate,
- ❑ procesul autonom trebuie să se desfășoare **în timp real**, deci trebuie adoptată o abordare de natură să permită identificarea promptă a căii către sursa de energie din rețeaua de distribuție a energiei electrice și

- regimul de funcționare rezultat în urma manevrelor să garanteze reușita RDEE prin (in)validarea manevrelor în ceea ce privește (supra-)încărcarea elementelor de rețea și încadrarea tensiunilor în bandă.

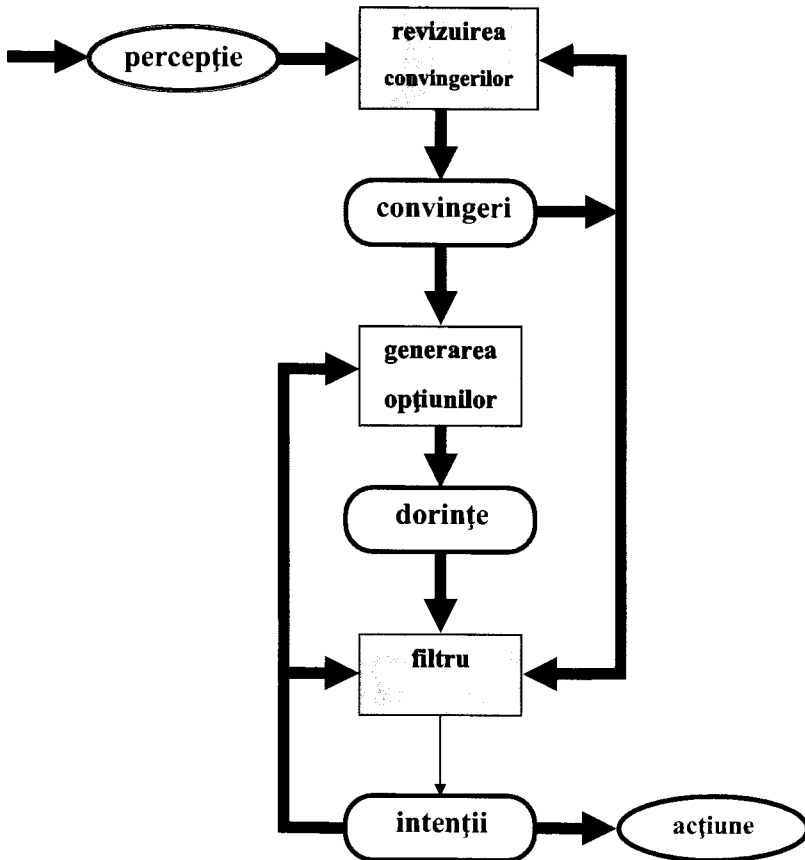
Când a apărut un incident în rețeaua de distribuție a energiei electrice, impunându-se RDEE, dispecerul procedează la lansarea SMARA prin intermediul stației sale de lucru acționând un buton de pe interfața om-mașină. Din acel moment, dispecerul și mașina comunică prin intermediul unor interfețe pentru fiecare mod de operare al SMARA:

5.3. Implementarea aplicației SMARA

Aplicația SMARA a fost concepută sub forma unui produs-program implementat pe stația de lucru a dispecerului. Aplicația este lansată ori de câte ori are nevoie dispecerul. Este legată la baza de informații prin intermediul procedurilor pe care furnizorul sistemului de monitorizare și control al distribuției energiei electrice le-a pus la dispoziția clientului, astfel încât SMARA percepe în timp real starea rețelei de distribuție a energiei electrice și comunică totodată aceste schimbări ale stărilor aparatelor de comutație primară atunci când manevrele sunt executate manual și confirmate prin interfața om-mașină.

Modelarea elementelor de rețea autonome și non-autonome, precum și a elementului autonom virtual de genul "dispecerului" s-a făcut prin intermediul unor clase de obiecte distincte. Pentru clasa de obiecte non-autonome s-au definit stările acestora, adică deschis, închis, conectat, deconectat, declanșat, "condamnat", supra-încărcat, et. Pentru clasele de obiecte autonome s-au definit în mod distinct pentru "dispecer", nodurile consumatoare și nodurile generatoare comportamentul BDI, specificându-se în mod strict "convingerile", "dorințele" și "intențiile" fiecărei clase.

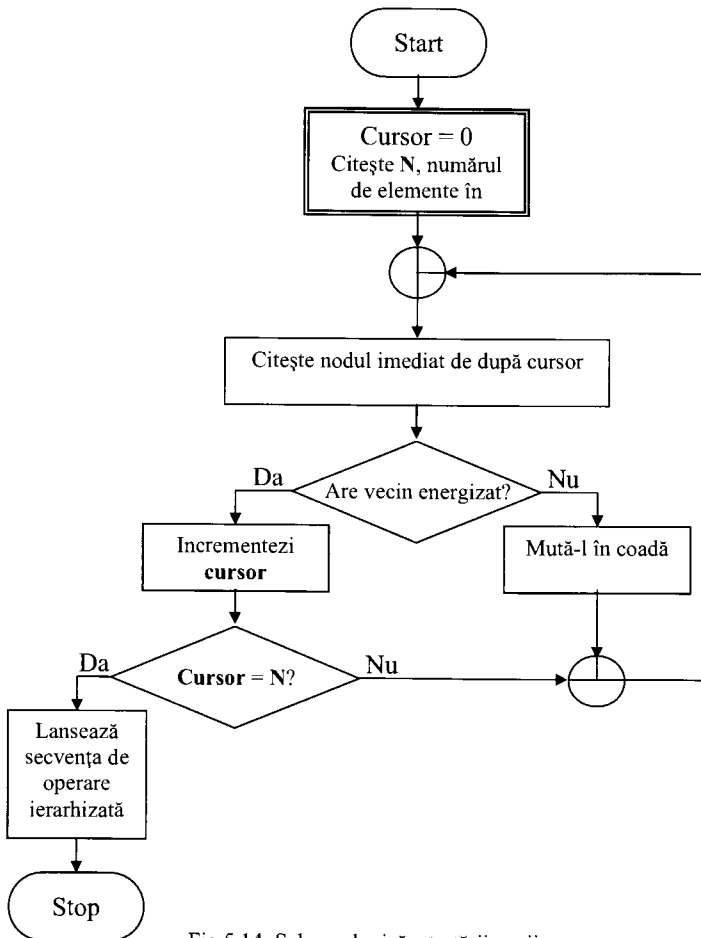
Comportamentul a fost modelat prin proceduri specifice fiecărei clase de obiecte autonome, în toate cazurile respectându-se schema logică ilustrată în figura nr.5.12, această schemă logică fiind în fapt copia unei arhitecturi BDI. În principiu, este aceeași pentru toate



- Fig.5.12: Schema logică a procedurii modelând comportamentul BDI -

clasele obiectelor autonome, dar diferă de la o clasă la alta, procedurile ținând seama de proprietățile fiecărei clase. În mod concret, "convingerile", "doriștele" și intențiile sunt descrise ca proprietăți pentru fiecare clasă de obiecte. Revizuirea "convingerilor", generarea "opțiunilor" și filtrarea acestora se face în mod diferit în procedura de lucru a fiecărei clase de obiecte autonome. În ceea ce privește percepțiile și acțiunile, în cadrul fiecăreia dintre procedurile amintite, sunt concepute structuri alternative de genul "select case" pentru a face alegerea între cele trei moduri de operare, auto, ierarhizată și arbitrată.

Procesul identificării și activării nodurilor-sursă alternative implică un "transfer de competență", adică luarea în considerare și a topologiei de la nivelul de tensiune superior, și



- Fig.5.14: Schema logică a tratării cozii -

căutarea căii de acces spre tensiune. Ca urmare, procedura va lua în considerare nivelul de tensiune superior și va căuta accesul "zonei calamitate" la tensiune. Pentru aceasta, nodurile-sursă dezenergizate de la nivelul de tensiune inferior apelează la căutarea tensiunii. Topologia de la nivelul de tensiune superior este un graf "auto-orientat" pornind de la sursele de tensiune ale acestuia. În mod similar secvenței de operare ierarhizată se caută calea spre sursă, dar condiția de validare nu este existența vecinătății unui nod energizat, ci validitatea căii

de acces spre nodul vecin, adică verificarea faptului că nu există elemente de rețea condamnate. Este același mod în care se caută starea închis / deschis a separatoarelor și conectat / deconectat a întrerupătoarelor pentru a putea determina manevrele de executat, indiferent de modul de operare. Același mod de lucru se poate aplica și la tratamentul cozii de așteptare cu număr prea mare de noduri. Căutarea nu se face prin cicluri imbricate cu număr mare de iterații deoarece s-ar pierde timp cu calculația, motiv pentru care am apelat la o nouă metodă, inspirată din tehnica localizării defectelor pe liniile electrice subterane.

5.4. Testarea aplicației SMARA

Când apare situația imperativă a RDEE, utilizatorul apelează SMARA pe butonul în cauză de pe interfața om-mașină a stației sale de lucru, ocazie cu care se afișează pe ecran interfața om-mașină a SMARA. Aceasta se prezintă sub forma foi de manevră, fiecare linie descriind o manevră de executat. Ordinea este cea logică. Manevrele care s-au executat sau vor urma a fi executate sunt afișate cu litere gri, dar manevra curentă fiind afișată prin caractere luminoase. Odată confirmată manevra, fie automat prin telecomandă sau monitorizare cu IEDs, fie manual prin apăsarea butonului afișând manevra curentă, butonul său se dezactivează și caracterele trec pe culoarea gri, activându-se butonul manevrei următoare, caracterele descriind manevra devenind luminoase.

Funcționarea SMARA este posibilă ca urmare a concepției, elaborării și realizării unor funcții specifice:

- transferului competenței gestionării operative** între nivelele de tensiune astfel încât, la apariția unor "zone calamitate" la nivelul inferior să se procedeze mai întâi la identificarea surselor ce trebuiesc activate de la nivelul superior.
- auto-orientarea** grafului modelînd rețeaua de distribuție a electricității, problemă dificilă în ceea ce privește analiza grafurilor; pornind de la premiza că informația ca atare este percepția schimbării formelor în mediul înconjurător, rezultă că imaginea percepută a realității este cu atât mai "fotografică" cu cât percepția este "atomizată" pînă la nivelul fiecărui nod autonom, astfel încît imaginea efectiv "fotografică" a realității să se agrege în mod coerent prin inter-comunicarea dintre noduri.
- restricția unică** permite constituirea cozii de așteptare.
- segmentarea vectorială** permite eludarea ciclurilor imbricate .

Capitolul 6

Direcții viitoare de cercetare pentru RDEE

Obiectul prezentului capitol este acela de a creiona direcții viitoare de cercetare și dezvoltare a aplicației SMARA astfel încât să răspundă cerințelor RDEE în contextul viitoarelor provocărilor ale distribuției energiei electrice.

6.1. Provocările sistemului de distribuție a energiei electrice

În prezent, distribuția electricității evoluează de la rețeaua **pasivă** având configurație radială și vehiculând electricitatea din rețeaua de transport către consumatori la rețeaua **activă** având configurație buclată și vehiculând către consumatori electricitatea preluată la "generarea distribuită", adică de la generatoarele eoliene, hidraulice, solare, etc. făcând injecția de putere direct în rețeaua de distribuție a electricității. Din această cauză apar o serie de probleme privind gestionarea distribuției electricității, probleme cărora li s-au găsit soluții momentane.

Esența provocărilor privind distribuția electricității privește următoarele aspecte:

- ☐ stadiul existent al rețelelor pasive de distribuție a electricității a impus adoptarea inteligenței artificiale ca soluție pentru elaborarea deciziilor în vederea lichidării avariilor,
- ☐ provocarea prezentă a apariției rețelelor active de distribuție a electricității ca urmare a apariției generării distribuite a impus integrarea noilor noduri autonome cu valențe DASf maximi gestionate de către SMARA și apariția noului modul pentru gestionarea încadrării tensiunilor în bandă, modul integrabil în aceeași aplicație SMARA, și
- ☐ provocarea apariției rețelelor virtuale de distribuție a energiei electrice ca urmare a apariției generării virtuale, ceea ce impune realizarea unor sisteme autonome dedicate, aceasta constituind o conjunctură stimulantă pentru dezvoltarea actualei aplicații SMARA..

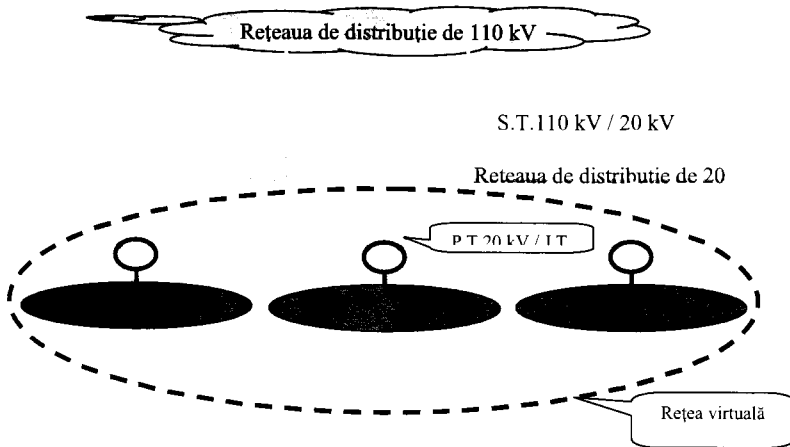
6.2. Rețele virtuale pentru distribuția energiei electrice

În prezent, se discută deja de *power smart networks*—"rețele electrice inteligente" și de *power virtual networks*—"rețele electrice virtuale", primele fiind, în ultimă analiză, o informatizare pînă la limita extremă a actualelor rețele de distribuție a energiei electrice și/sau a rețelelor-proprietar de consum a energiei electrice, celelalte fiind rețele de

distribuție a energiei electrice pentru care se pune problema coordonării prin decizii care să eficientizeze la maximum posibil rețelele active. În consecință, tocmai aceste "rețele virtuale", prin faptul că implică elaborarea deciziilor complexe și totodată prompte în execuție, au drept soluție implementarea corespunzătoare a sistemelor autonome de monitorizare și control.

"Rețeaua virtuală" se definește prin luarea în considerare a următoarelor nivele de agregare:

- inferior—generatoarele virtuale și aparatele de încărcare și stocare a energiei (baterii de acumulare, redresoare, invertoare, întrerupătoare, separatoare, etc.).
- intermediar—P.T.-J.T./ M.T. de interconectare a rețelei virtuale locale cu rețeaua de distribuție de M.T și
- superior—ansamblul subrețelelor virtuale interconectate cu rețeaua de distribuție de M.T.—vezi fig.nr.6.6.



- Figura nr.6.6: interconectarea "sub-rețelelor virtuale" în "rețeaua virtuală" -

Concluzii

Prezenta lucrare a abordat o problema importanta din cadrul conducerii sistemelor de distributie a energiei electrice si anume *problema restabilirii* acestui sistem în urma producerii unei avarii. Din punctul de vedere al teoriei conducerii sistemelor automate, problematica aceasta se încadreaza în studiul sistemelor autonome de control (comanda), unde obiectivul de conducere, gasirea unei noi structuri a RDEE care sa energizeze zonele avariate, este o "referinta" generica, obiectivul fiind atins printr-o functionare autonoma, pe baza unor informatii locale, cu senzori locali si elemente de executie locale.

Natura problemei restabilirii dupa avarie, implica un SAC cu mai multe elemente de comanda autonome si cu caracteristicile agentilor inteligenti. Practic, modalitatea de implementare a SAC este realizarea unui SMA, unde agentii sunt asociati magistralelor de transport a energiei electrice (barelor sau nodurilor, cum au mai fost numite în lucrare). De aceea, au fost folosite cunostinte dintr-un alt domeniu stiintific, cel al inteligentei distribuite, mai precis al SMA. Asa cum un regulator se poate implementa printr-un program, tot asa SAC utilizat s-a implementat printr-un SMA.

Au fost prezentate doua SMA cu arhitecturi diferite din motive explicate în Cap. 5. Primul a fost implementat si simulat în mediul de dezvoltare JACK, iar cel de-al doilea SMARA a fost dezvoltat ca o aplicatie JAVA ce va putea integra noi functionalitati. Aplicatia SMARA a fost testata, în regim simulat, pe doua studii de caz reale, de complexitate ridicata: unul pentru un segment din retea de inalta tensiune si celalalt pentru o portiune din retea de medie tensiune a municipiului Galati. Testele au demonstrat ca abordarea este nu numai realista, dar si foarte utila, putandu-se imagina deja o utilizare în ghid operator.

Principalele contributii ale autorului prezentei lucrari sunt urmatoarele:

- adaptarea structurii SMA cu autonomie completa prezentata în Cap. 3 (vezi [94]) la mediul de dezvoltare JACK;
- proiectarea elementelor ce realizeaza comunicarea agentilor si negocierea în spiritul arhitecturii BDI implementata în JACK;
- definirea evenimentelor, mesajelor, planurilor de tratare a evenimentelor, datelor ce medeaza convingerile, dorintele si intentiile în concordanta cu o rezolvare iterativa si cooperanta a probleme restabilirii;
- conceperea unei sesiuni de simulare si testare a evolutiei SMA;

Cea de a doua soluție de SMA cu autonomie supervizată a prilejuit autorului posibilitatea de a avea noi contribuții:

- considerarea unui agent virtual având rolul de "dispecer". În felul acesta se poate fructifica experiența din dispecerate în rezolvarea avariei;

- conceperea a trei moduri de operare pentru nodurile RDEE:

a). **auto-operarea** (nodul acționează exclusiv din proprie inițiativă prin apelarea ajutorului unui vecin energizat).

b).operarea **ierarhizată**; În absența unui vecin energizat, nodul, ca agent autonom, se adresează agentului virtual "dispecer" care procesează coada de așteptare a nodurilor care și-au anunțat starea de dificultate astfel încât să o ordoneze într-un mod în care se lansează o succesiune de secvențe de auto-operare în mod ordonat:

- autoorientarea grafului pe baza surselor de electricitate,
- identificarea grafului valid în funcție de "condamnarea" aparatelor de comutație primară ca urmare a apariției defectelor ireversibile,
- identificarea instanțelor de căi de acces la sursele de electricitate în cadrul grafului valid procesând vectorii prin ordonarea căilor de acces astfel încât să se asigure condițiile de funcționare corectă a rețelei de distribuție a electricității, identificarea topologiei finale a rețelei de distribuție a electricității și procesarea ordonării căilor de acces viabile și ale identificării topologiei optimele cu ajutorul tehnicilor de intercomunicare gen *send by receive* între agenții autonomi și ordonare a vectorilor în mod similar localizării defectului pe un cablu electric, astfel încât procesarea să se producă instantaneu și să năntirzie din cauza unor structuri ciclice imbricate și folosind vectori foarte mari,
- ordonarea cozii de așteptare alterând urgența nodului în funcție de vecinătatea sursei de electricitate
- transferul secvențial de competență către nodurile din coada de așteptare astfel încât să aibă loc o succesiune de autooperări reușite pentru fiecare nod în parte

c). operarea **arbitrată**; În cazul în care unul sau mai multe noduri în dificultate sunt în afara grafului valid, are loc un transfer de competență în urma căruia "dispecerul" conduce acțiunea:

- caută surse "reci" de electricitate la nivelul de tensiune superior astfel încât să integreze în mod virtual nodul aflat în dificultate într-un graf valid,
- aducerea în stare "caldă" a surselor de electricitate identificate
- transferul de competență de la nivelul de tensiune superior la cel inferior (acolo unde s-a produs avaria) astfel încât "dispecerul" să treacă de la modul de operare arbitrat la modul de operare ierarhizat.

- conceperea unei sesiuni de simulare și testare a evoluției SMA, în cadrul aplicației SMARA

Pe lângă fundamentarea teoretică și definirea tehnică a agenților autonomi și comportamentului acestora, doctorandul a elaborat și aplicația soft necesară în vederea realizării practice a contribuției sale.

În acest sens, s-a interconectat virtual cu sistemul SCADA & DMS al unei companii distribuitoare de electricitate aplicația elaborată de către doctorand. Aplicația este

Referințe bibliografice

- [1] Fogel DB, Fukuda T, Guan L (1999) Scanning the Special Issue – Technology on Computational Intelligence. IEEE Proceedings Special Issue on Computational Intelligence, vol 87, n°9
- [2] Antsaklis PJ, Passino KM (1993) An Introduction to Intelligent and Autonomous Control
- [3] Abel E, e.a. (1993) A multi-agent approach to analyze disturbances in electrical networks. In Proceedings of the 4th ESAP Conference, Melbourne, Australia, pp 606÷611
- [4] Talukdar S (1993) Asynchronous Teams, Proceedings of the 4th ESAP Conference, Melbourne, Australia, pp 647÷655
- [5] Roberts DA (2004) Electricity Distribution Network Active System, © SP Power Systems Ltd, ScottishPower Plc
- [6] Rehtanz Ch (2003) Autonomous Systems and Intelligent Agents in Power System Control and Operation, Springer-Verlag Berlin
- [7] Fleischmann A (1994) Distributed Systems – Software, Design & Implementation, Springer-Verlag Berlin
- [8] International Organization for Standardization (1984) Basic Reference Model for Open Systems Interconnection, ISO 8072
- [9] Miller A (1994) From here to ATM, IEEE Spectrum, June, pp 20÷24
- [10] ISO / IEC 9506 (1990) Manufacturing Message Specification
- [11] IEC 61870: Telecontrol equipment and systems – Part 6: Telecontrol protocols compatible with ISO standards and ITU-T recommendations, 2001 / 2001
- [12] Draft IEC 61850: Communication networks and systems in substations, Nov 2002
- [13] Kostic T, Preiss O, Frei Ch, Towards the Formal Integration of Two Upcoming Standards: IEC 61970 and IEC 61850, Proceedings of LESCOPE 2003 Conference, Montreal, 7÷9 May, 2003
- [14] Draft IEC 61970: Energy Management System Application Programming Interface, Oct 2003
- [15] OMG Unified Modelling Language Specification, Version 1.4, Sep 2001
- [16] Draft IEC 61968 System Interfaces for Distribution Management, May 2001
- [17] Öszu MT, Valduriez P (1999) Principles of Distributed Database Systems, 2nd Edition, Prentice Hall, New Jersey
- [18] Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison Wesley

- [19] Rehtanz Ch (2003) *Autonomous Systems and Intelligent Agents in Power System Control and Operation*, Springer-Verlag Berlin
- [20] Yao X (1999) *Evolving Artificial Neural Networks*. IEEE Proceedings, vol 87, n^o9
- [21] Kiendl H (1998) *Self-Organizing Adaptive Moment-Based Clustering*. IEEE International Conference on Fuzzy Systems, Anchorage, Alaska, IEEE Press, Piscataway, NJ, vol 2, pp 1470÷1475
- [22] Martinez T, Schulten K (1991) A 'neural gas' network learns topologies. In Kohonen T et al, *Artificial Neuronal Networks*, North Holland, Amsterdam, vol I, pp 397÷402
- [23] Fritzke B (1995) A 'growing neuronal gas' network learns topologies. *Advances in Neuronal Information Processing Systems 7*, MIT Press. Cambridge
- [24] Shafer G (1976) *A mathematical theory of evidence*. Princeton University Press. London
- [25] Dempster A (1967) Upper and lower probabilities induces by a multivalued mapping. *Annals of Mathematical Statistics*. vol 38. pp 325–339
- [26] Rosenschein JS, Zlotkin G (1994) *Designing Convention for Automated Negotiation*. AI Magazine, pp 29÷46
- [27] Wooldridge M (1999) *Intelligent Agents*, The MIT Press
- [28] Wooldridge M, Jennings NR (1995) *Agent Theories, Architectures and Languages: A Survey*
- [29] Weiss G (2000) *multi-agent Systems – A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, Massachusetts
- [30] Brooks RA: *A Robust Layered Control System for a Mobile Robot*, IEEE Journal of Robotics and Automation, RA-2, April, 12÷23, 1986
- [31] Brooks RA (1991) *Intelligence without Reason*, Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia, pp 569÷595
- [32] Brooks RA: *Subsumption Architecture Background*,
http://ai.eecs.umich.edu/cogarch3/Brooks/Brooks_Background.html
- [33] Kowalski R, Sadri F (1997) *An Agent Architecture that Unifies Rationality with Reactivity*, Department of Computing, Imperial College, London, UK,
<http://citeseer.nj.nec.com/kowalski97agent.html>
- [34] Rao AS, Georgeff MP (1991) *Modelling rational agents with a BDI-architecture*, Proceedings of the 2nd International Conference on Principles of Knowledge, Representation and Reasoning, Morgan Kaufmann Publishers, pp 473÷484
- [35] Norman T (1995) *Belief, desire and intention architectures*,
<http://www.esd.abdn.ac.uk/~tnorman/publications/atal95/bdi.html>

- [36] Cohen P, Levesque H (1990) Intention is Choise with Commitment, *Artificial Intelligence*, 42
- [37] Kinny D, Georgeff M (1991) Commitment and Effectiveness of Situated Agents, *Intelligence without Reason, Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, pp 82÷88
- [38] Wooldridge M (1999) *Inteligent Agents*, The MIT Press
- [39] Gulliver (1979) *Disputes and Negotiations*, Academic Press, New York
- [40] Chankong V, Haimes Y (1983) *Multi-objective Decision Making Theory and Methodology*, North-Holland
- [41] Hobbs B, Meier P (2000) *Energy decisions and the environment: A guide to the use of multi-criteria methods*, Kluwer, Boston, Massachussets
- [42] Bernoulli D (1954) *Exposition of a New Theory of the Measurement of Risk*, English translation: *Econometrica*
- [43] Von Neumann J, Morgenstern O (1944) *Theory of Games and Economic Behavior*, John Wiley, New York
- [44] Keeney R, Raiffa H (1976) *Decisions with Multiple Objective: Preferences and Value Tradeoffs*, John Wiley, New York
- [45] Fishburn P (1982) *The Foundations of Expected Utility*, D.Reidel Publishing Company
- [46] Fishburn P (1988) *Non-linear Preference and Utility Theory*, the *John Hopkins University Press*
- [47] Tapan B (1997) *Decision-making under uncertainty*, Saint Martin's Press, New York
- [48] Strauss A (1978) *Negotiations: Varieties, Contexts, Processes and Social Order*, Jossey-Bass Publishers, San Francisco
- [49] Raiffa H (1982) *The art and science of negotiation*, Harvard UniversitzPress, Cambridge, Massachussttes
- [50] Rubenstein A (1982) Perfect echilibrium in a bargaining model, *Econometrica* 50, pp 97÷109
- [51] Kraus S (2001) *Strategic Negotiation in Multi-agent Environment*, the MIT Press, Cambridge, Massachussets
- [52] Rosenschein J, Zlotkin G (1994) *Rules of Encounter*, the MIT Press, Cambridge, Massachussets
- [53] Ehtamo H, Hamalainen R (2001) Interactive Multiple-Criteria Methods for reaching Pareto Optimal Agreements in Negotiations, *Journal of Group Decisions and Negotiations*, vol 10, pp 475÷491

- [54] Walton R, McKersie R (1965) A Behavioral Theory of Labor Negotiations, McGraw-Hill, New York
- [55] Foundation of Intelligent Physical Agents (FIPA) specifications available at <http://www.fipa.org>
- [56] Bărbuceanu M, Fox MS (1999) COOL: A language for describing coordination in multi-agent domains, in First International Conference on Multi-Agent systems, AAAI Press / MIT Press, pp 17÷24
- [57] Smith R (1980) The Contract Net Protocol: High Level Communication and Control in Distributed Problem Solver, IEEE Transmission on Computers, col C-29, n°12, pp 1.104÷1.113
- [58] Smith R, David R (1983) Negotiation as a metaphor for distributed problem solving, Artificial Intelligence 20, pp 60÷109
- [59] Faratin P, Sierra C, Jennings N (1997) Negotiation decision functions for autonomous agents, International Journal of Robotics and Autonomous Systems, vol 24, n°3÷4, pp 159÷182
- [60] Vishvanathan V, McCalley J, Honavar V (2001) A multi-agent system infrastructure and negotiation framework for electric power systems, Proceedings of IEEE Porto Power Technology Conference, Porto, Portugal
- [61] Hogg LM, Jennings NR (1997) Socially Rational Agents – Some Preliminary Thoughts, Proceedings of the 2nd Workshop on Practical Reasoning and Rationality, Manchester, pp 160÷162
- [62] Hogg LM, Jennings NR (1997) Socially Rational Agents, Proceedings of AAAI Fall Symposium on Socially Intelligent Agents, pp 61÷63
- [63] Weiss G (1996) Ecai-96 Workshop on Learning in Distributed Artificial Intelligence
- [64] Decker KS (1987) Distributed problem solving: A survey, IEEE Transactions on Systems, Man and Cybernetics, vol 17(5), pp 729÷740
- [65] Van Dyke Parunak H (1996) Applications of Distributed Artificial Intelligence in Industry, in O'Hare GMP, Jennings NR: Foundation on Distributed Artificial Intelligence, Willy Interscience, pp 139÷164
- [66] Stone P, Veloso M (1997) Multi-agent Systems: A Survey from a Machine Learning Perspective, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania,
http://www-2.es.cmu.edu/afs/es/usr/pstone/public/papers_97MAS-survey/revise-survey.html

- [82]Ferber J. *Le systemes multi-agent . Vers une intelligence collective*. Ed. InterEditions. Paris, 1995
- [83]Friedman-Hill E.J., *Jess, The Java Expert System Shell*,
<http://heerzberg.ca.sandia.gov/jess>, version 6.1a5, 15th January 2003
- [84]Gasser *Social conceptions of knowledge and action:DAI foundations and open systems dynamics*, 1998
- [85]Hayes-Roth B.: *An Architecture for Adaptive Intelligent Systems, Artificial Intelligence: Special Issue on Agents and Interactivity*, 72, 329-365, in [Franklin], 1995
- [86]Hossack J.H., Burt G.M., McDonald J.R., Cumming T., and Stoke J., *Progressive Power System Data Interpretation and Information Dissemination*", in Proc. 2001 IEEE Transmission and Distribution Conference and Exposition, 2001
- [87]Jennings N.R., *On agent-based software engineering. Artificial Intelligence*, 2000
- [88]Jung J., Liu C-C., Hong M., Gallanti M., Tornielli G., *Multiple Hypotheses and Their Credibility in On-Line Fault Diagnosis*, IEEE Transactions on Power Delivery, v16, no 2, pp 225-230, 2002
- [89]Kezunovic M., Rikalo I., Vesovic B., Goiffon S.L., *The Next Generation System for Automated DFR File Classification, Fault & Disturbance Analysis Conference*, May 1998.
- [90]Kinny D., Georgeff M.: *Commitment and Effectiveness of Situated Agents*, Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91), 82-88, Sydney, Australia, 1991, in [Weiß 2000]
- [91]Krishna V., Ramesh V.C., *Intelligent Agent for Negotiation in Market Games*, Part I: Model.IEEE Trans. on Power System, vol 13, no 3, pp 1103-1108, 1998
- [92]Krishna V., Ramesh V.C., *Intelligent Agent for Negotiation in Market Games*, Part II: Application.IEEE Trans. on Power System, vol 13, no 3, pp 1109-1114, 1998
- [93]Maes P.: *Artificial Life Meets Entertainment: Life like Autonomous Agents*, Communications of the ACM, 38, 11, 108-114, 1995, in [3S. Franklin]
- [94]Nagata T., Sasaki H., *Multi-Agent Approach to Power System Restoration*. IEEE Transaction on Power system, vol 17, no 2, pp 457-462, 2002
- [95]Nwana H. S.: *Software Agents: An Overview*, <http://www.labs.bt.com/projects/agents/publish/papers/review2.htm>, 1996
- [96]Russell S. J., Stuart J., Norvig P.: *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, NJ, Prentice Hall, 1995
- [97]Shoham: *Agent-oriented programming*, 1998

- [98] Talukar S., Ramesh V.C: *A Multi-agent Technique for Contingency Constrained Optimal Power Flows*, IEEE Trans. On Power System, vol 9, no 2, 855-861, 1994
- [99] Tecuci Gh.: *Agent-oriented programming*, 1998
- [100] Vale Z.A., and Fernandes M.F., *Better KBS for real-time applications in power system control centers: the experience of the SPARSE project*, Computers in Industry, vol. 37, pp. 97 –111, 1998.
- [101] Weiß G. 2000 (ed.): *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, Massachusetts, 2000
- [102] Wooldridge M., et al, *Intelligent Agents: Theory and Practice*, The Knowledge Engineering Review, vol. 10, n 2, pp. 115-152, 1995
- [103] Wooldridge M. and Jennings N.R: *Agent Theories, Architectures, and Languages: A Survey*, 1995
- [104] Wooldridge M.: *Intelligent Agents*, The MIT Press, 1999
- [105] Phil Taylor, Liana Cipcigan, 2006, "Small scale Energy Zones for Effective Participation of SSEGs in Energy Market", *WEC Regional Forum—FOREN 2006*, ISBN 973-720-032-2.
- [106] "Consideration of reference impedances and public supply network impedances for use in determining disturbance characteristics of electrical equipment having rated current= <75 A per phase", *IEC Technical Report 60725*, ANSI.
- [107] Steve Ingram, Sarah Probert. 2003, "The Impact of Small Scale Embedded Generation in the Operating Parameters of Distribution Networks", *D.B.Power BTI New and Renewable Energy Programme*.
- [108] Bumbz, JR and Stannard N., 2006, "Performance aspects of mains connected small scale wind turbines", Submitted to *Proceedings IEE-Electrical Power Applications*.
- [109] J.A.Lopes, 2002, "Integration of dispersed generation on distribution networks – impact studies", *IEEE Power Engineering Society Winter Meeting*.

