Adrian CURAJ

CONDUCEREA SISTEMELOR DE FABRICAŢIE INTEGRATE ÎN ARHITECTURI DE ÎNTREPRINDERE VIRTUALĂ

EDITURA DIDACTICĂ ŞI PEDAGOGICĂ
Bucureşti, 2000

© 1999. Toate drepturile asupra acestei ediţii
sunt rezervate Editurii Didactice şi Pedagogice

Adresa : EDITURA DIDACTICA SI PEDAGOGICA

Str. Spiru Haret 12, sector 1
Bucuresti 70738
Telefon: 315 3820, 313 3470

Redactor: Mirela CONSTANTIN
Tehnoredactor: Octav DRAGAN

Coperta:

Bun de tipar: 30.11.1999 Format: 16/70x100.

ISBN: 973-30-9925-2

Imprimat în Romania
Tipografia: BREN

CONDUCEREA SISTEMELOR DE FABRICAŢIE INTEGRATE ÎN ARHITECTURI DE ÎNTREPRINDERE VIRTUALĂ

MANAGING MANUFACTURING SYSTEMS INTEGRATED IN VIRTUAL ENTERPRISE ARCHITECTURES

PREFAŢĂ

Această lucrare se adresează unui public larg şi sper să găsească o bună audienţă. Ea îşi propune să fie un ghid pentru cei care sunt interesaţi de domeniul conducerii sistemelor de fabricaţie şi abordează în special problema conducerii sistemelor de fabricaţie integrate într-o structură de întreprindere virtuală. Entităţile sistemului întreprindere virtuală, denumite fabricatori, sunt agenţi autonomi care negociază formarea unor alianţe temporare având ca obiective utilizarea în comun de resurse şi cunoştinţe complementare, precum şi creşterea vitezei de reacţie faţă de oferta de afaceri.

Au fost definite noţiuni fundamentale care stau la baza construcţiei acestei lucrări. Structurarea s-a dorit a fi cît mai logică cu putintă, intenţia fiind ca lucrarea sa fie utilă atât studenţilor cît şi celor care cercetează în domeniu.

Lucrarea constituie un demers original, oferind un punct de vedere coerent asupra unei probleme de actualitate şi în plină evoluţie - noile paradigme de conducere a sistemelor de fabricaţie în societatea informaţională.

Conducerea întreprinderii virtuale, în toate etapele ciclului de viată, este formulată ca o problemă de conducere de sistem în timp real şi se fac o serie de consideraţii referitoare la universul de timp şi constrângerile ce apar. Corespunzător tipului de coordonare a întreprinderii virtuale, se propune o arhitectură descentralizată bazată pe agenţi autonomi şi o arhitectură bazată pe agenţi autonomi coordonaţi central, care se regăsesc în situaţia specifică corporaţiilor şi organizării companiilor pe centre de profit.

Lucrarea propune un model generic de sistem de conducere avînd la bază trei module: planificare proactivă, negociere şi control reactiv. La nivel de fabricator, pentru conducerea fabricii fizice, se propune utilizarea unei arhitecturi de control supervizor şi achiziţie de date. Studiul de caz propune o arhitectură de pilot de întreprindere virtuală/laborator virtual, format din trei entităţi/laboratoare independente din mediul universitar, şi prezintă diverse soluţii de sistem de conducere, testate în aceste laboratoare.

Se cuvine să mulţumesc şi pe această cale celor care de-a lungul anilor m-au îndrumat atât în calitate de dascăli cît şi de colegi. Cu regretul că nu îi voi putea aminti pe toţi, multumesc domnilor: prof. univ. dr. ing. Ioan Dumitrache, prof. univ. dr. ing. Aurelian Mihai Stănescu şi prof. univ. dr. ing. Radu Dobrescu.
Trebuie să mulţumesc şi celorlalţi colegi de catedră şi nu în ultimul rînd sefului Catedrei Automatică şi Informatică Industrială a Facultăţii de Automatică şi Calculatoare din Universitatea Politehnica din Bucuresti, prof. univ. dr. ing. Traian Ionescu, pentru mediul creat şi întelegera acordată de-a lungul timpului.

Bucureşti, 1 decembrie 1999 *Autorul*

CUPRINS

1. Integrarea fabricaţiei prin calculator-CIM
1.1. Scurt istoric al evoluţiei sistemelor de fabricaţie 9
1.2. Întreprinderea industrială cu fabricaţie discretă 10
1.2.1. Întreprinderea ca sistem cu învăţare 12
1.2.2. Modelul clasic de organizare a fabricii 15
1.3. Integrarea fabricaţiei prin calculator 17
1.3.1. Platforme funcţionale pentru integrare activităţi 18
1.3.2. Arhitecturi de referinţă CIM 25
1.3.3. Ciclul de viaţă al întreprinderii CIM 27
1.3.4. Planificare şi control ierarhic 29
1.3.5. Arhitectura ierarhică de control 30
1.3.6. Model de sistem de control al activităţilor de producţie 33
1.3.7. Arhitectura de referinţă CIM-OSA 35
1.3.8. Avantaje şi limitări ale paradigmei CIM 39
Concluzii 43

2. Sistemul de fabricaţie post-CIM 45
2.1. Cerinţe pentru sistemul viitor de fabricaţie 45
2.2. Vectori de schimbare 48
2.2.1. Modelul sistemului holonic de fabricaţie 48
2.2.2. Model de sistem fractal de fabricaţie 56
2.2.3. Model de sistem bionic de fabricaţie 62
2.2.4. Comparaţie între modelele holonic/fractal/bionic 64
2.3 Flexibilitatea sistemelor de fabricaţie 67
Concluzii 70

3. Fabrica virtuală 73
3.1. Modelul fabricii virtuale 73
3.1.1. Arhitectura de fabrică virtuală 75
3.1.2. Obiecte, procese, agenţi în universul fabricii virtuale 78
3.1.3. Problema alocării activităţilor 80
3.1.4. Definirea noţiunii de fabricator 83
3.2. Platforma multi–dimensională pentru modelare 85
3.2.1. Dimensiunea–reprezentare funcţională 88
3.2.2. Dimensiunea–reprezentare structurală 93

3.2.3. Dimensiunea–reprezentare comportamentală 95
3.2.4. Dimensiunea–reprezentare contextuală 104
Concluzii 109

4. Întreprinderea virtuală 113
4.1. Sistemul întreprindere virtuală 113
4.1.1. Arhitectura de referinţă pentru întreprinderea virtuală 115
4.1.2. Interfaţă neutrală pentru integrarea partenerilor în întreprinderea virtuală 119
4.2. Proiectarea de alianţe tip întreprindere virtuală, bazată pe module fabricator 128
4.2.1. Reţea globală de cunoştinţe de reinginerie 128
4.2.2. Suport formal pentru proiectarea de alianţe tip întreprindere virtuală 130
4.2.3. Bibliotecă cu acces inteligent la cunoştinţe de reinginerie 139
4.3. Problema conducerii sistemelor de fabricaţie integrate în structuri de întreprindere virtuală 143
4.4. Arhitectură de control supervizor şi achiziţie de date pentru fabricator 148
Concluzii 152

5. Studiu de caz: FABRICATOR–pilot de laborator virtual 155
5.1. Arhitectura de sistem 155
5.1.1. Laborator pentru cercetări în domeniul fabricaţiei inteligente 159
5.1.2. Laborator de cercetare pentru SED şi SCADA 175
5.1.3. Laborator pentru cercetări în domeniul tehnologiei mecanice şi a managementului industrial 176
Concluzii 178

Bibliografie 189

Anexe 187

1

INTEGRAREA FABRICAŢIEI PRIN CALCULATOR-CIM

1.1. SCURT ISTORIC AL EVOLUŢIEI SISTEMELOR DE FABRICAŢIEI

Evoluţia sistemelor de fabricaţie este jalonată de definirea unor modele şi soluţii de referinţă, care încapsulează experienţa şi nivelul tehnologic specifice perioadei respective. Conţinutul şi convergenţa acestor avansuri manageriale, tehnologice şi organizatorice sunt sintetizate în *paradigme*, destinate să clarifice, să orienteze şi să compatibilizeze eforturile de perfecţionare a sistemelor de producţie într-o perioadă dată. Paradigma este definită [Jovane,94] ca fiind rezultatul interacţiunii între inovaţia tehnologică, tehnologia informaţiei, cerinţele pieţii şi evoluţia contextului socio-economic. În conformitate cu [FitzGerald,87] o paradigmă se poate defini ca o modalitate de a gândi în legătură cu o anumită problemă. În literatura de specialitate domeniul producţiei industriale discrete are două radăcini lingvistice. Referinţele din limba engleză folosesc termenul de MANUFACTURING, în timp ce referinţele din limba franceză utilizează pe cel de PRODUCTIQUE. În acest studiu s-a utilizat denumirea de „SISTEM DE FABRICAŢIE" care înglobează cele trei niveluti de activitate dintr-o întreprindere industrială, *fabricaţia* propriu-zisă, *logistica* şi *decizia* (fig.1.1.).

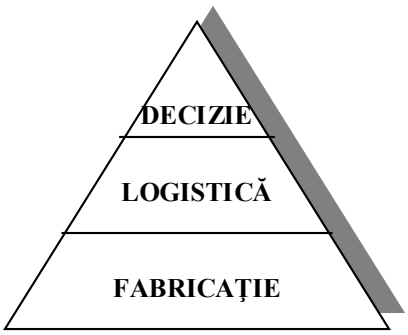


Fig. 1.1. Ierarhizarea nivelurilor într-un sistem de fabricaţie

Anii '60 au fost dominaţi de paradigma *liniei de transfer* ca soluţie de automatizare, în condiţiile în care productivitatea şi producţia de masă reprezentau principalele criterii de performanţă.

Prima parte a deceniului opt este caracterizată de paradigma *insulelor de fabricaţie*, care viza îmbogăţirea conţinutului profesional al diverselor ocupatii (job-uri), dar care implica importante creşteri ale costurilor de producţie. Spre sfârşitul deceniului a început să se impună modelul *automatizării flexibile*, orientat pe reducerea timpului de reacţie la cerinţele pieţei prin flexibilizarea soluţiilor de automatizare. *Este prima paradigmă care pune pregnant în evidenţă rolul informaticii în perfecţionarea sistemelor de producţie.*

La începutul anilor '90 modelul *fabricaţiei integrate cu calculatorul* urmăreşte să valorifice progresele rapide în domeniul tehnologiei informatice (cu familiarul acronim IT) pentru integrarea activităţilor de bază ale întreprinderii*: proiectare produse, proiectare procese, producţie*.

Noile paradigme de modelare, „*lean production"* (orientată pe adaptarea rapidă la cerinţele pieţii pe baza utilizării conceptului „*just în time*") şi *„human-centered"(*orientat pe activitatea de grup care acţionează purtând întreaga responsabilitate, dar având şi autonomia necesară realizării sarcinilor primite) subliniază noile tendinţe privind concepţia evoluţiei sistemelor de fabricaţie.

Acest ultim deceniu se caracterizează printr-o atenţie deosebită acordată modelării întreprinderii, sprijinită de evoluţia rapidă a suportului IT şi a comunicaţiilor, precum şi de orientarea spre colaborare şi dezvoltare de baze globale de cunoştinţe. Eforturile sunt concentrate pe echilibrarea diverselor orientări şi soluţii anterioare. Se propune conceptul de *automatizare balansată* [Camarinha'95], bazat pe abordarea concertată, sistemică a tuturor factorilor (economici, tehnologici şi antropocentrici) care pot influenţa rentabilitatea unei întreprinderi.

1.2. ÎNTREPRINDEREA INDUSTRIALĂ CU FABRICAŢIE DISCRETĂ

Figura 1.2.oferă o reprezentare globală de sistem pentru întreprinderea industrială. Întreprinderea ca sistem deschis urmăreşte permanent armonizarea sistemului de fabricaţie cu cerinţele mediului extern. Presiunea este dată de piaţă şi de schimbările tehnologice. Salturile majore au fost făcute odată cu evoluţia suportului informatic şi a automatizării flexibile. Modele pentru întreprinderea industrială sunt legate de nevoia de a orienta eforturile de proiectare a noii întreprinderi, sau de adaptare a unei întreprinderi deja existente la noul context apărut .

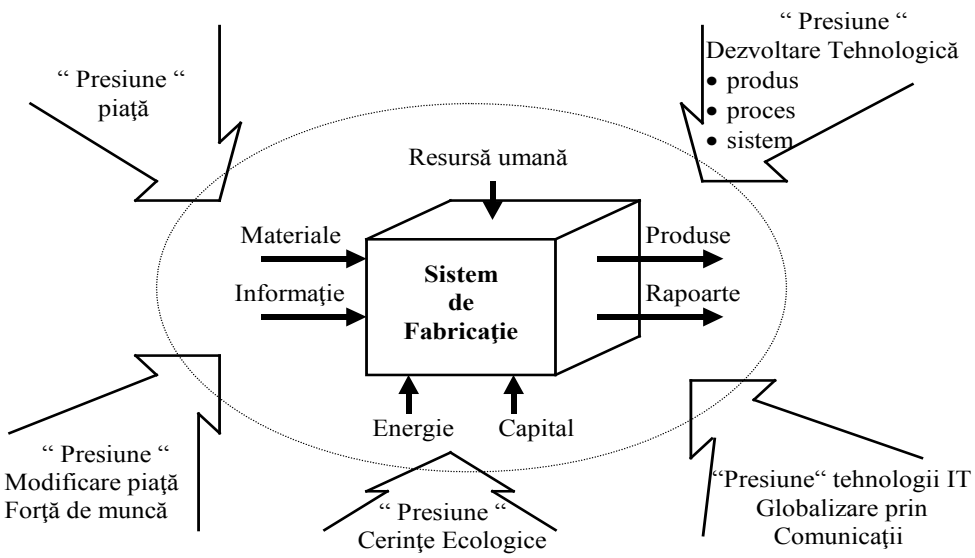


Fig.1.2. Sistemul întreprindere de fabricaţie

Dacă întreprinderea este văzută ca un sistem aflat într-o permanentă “căutare” a soluţiei optime de atingere a *obiectivelor* sale de afaceri, modelul sistemului de fabricaţie integrat prin computer (CIM–Computer Integrated Manufacturing) ca model global de sistem, susţinut de globalizarea pieţii, se constituie ca tendinţă ce sprijină efortul de apropiere de conceptul de „Fabrică a Viitorului*”* (*Factory of Future–FOF).*

Un sistem de fabricaţie are ca obiectiv major răspunsul rapid orientat cerinţă piaţă/client, care presupune :

- *creşterea flexibiliăţii (concept major);*
- *asigurarea calităţii constant ridicate;*
- *micşorarea ciclului de fabricaţie pe produs;*
- *micşorarea costurilor de producţie.*

Criteriul global de performanţă a unei întreprinderi industriale se poate formula astfel: *“Obţinerea unui număr cît mai diversificat de produse de calitate, cu costuri minime şi la momente de timp prestabilite, ţinând cont de restricţiile privind utilizarea cât mai eficientă a resurselor disponibile“.*

Factorii cost/calitate/termen de livrare se constituie în triunghiul forţelor de echilibru dinamic, care urmează să susţină un produs pe piaţă, de-a lungul întregului său ciclu de viaţă.

1.2.1. ÎNTREPRINDEREA CA SISTEM CU ÎNVĂŢARE

Teoria sistemelor deschise accentuează relaţiile strânse dintre sistem şi mediul în care acesta există şi evoluează. Aceste relaţii sistem-mediu sunt considerate esenţiale în adoptarea efectivă a deciziei.

Proiectarea unui sistem de fabricaţie necesită rezolvarea unei probleme de sistem complex, neliniar despre care există informaţii parţiale referitoare la starea iniţială . Ca urmare efortul de proiectare poate fi sprijinit în mod limitat de un număr mare de abordări specifice modelării liniare, care nu pot însă oferi soluţii de sistem, în majoritatea situaţiilor neputându-se face o predicţie a stării viitoare pornind de la starea curentă a sistemului.

Dându-se sistemul neliniar de o astfel de complexitate este imposibil să se stabilească cu certitudine că o intervenţie particulară asupra sa, îi va determina o traiectorie de stare având ca punct ţintă o stare viitoare estimată. În asemenea condiţii este mult mai realistă *transformarea întreprinderii într-un „sistem cu învăţare“,* figura 1.3.

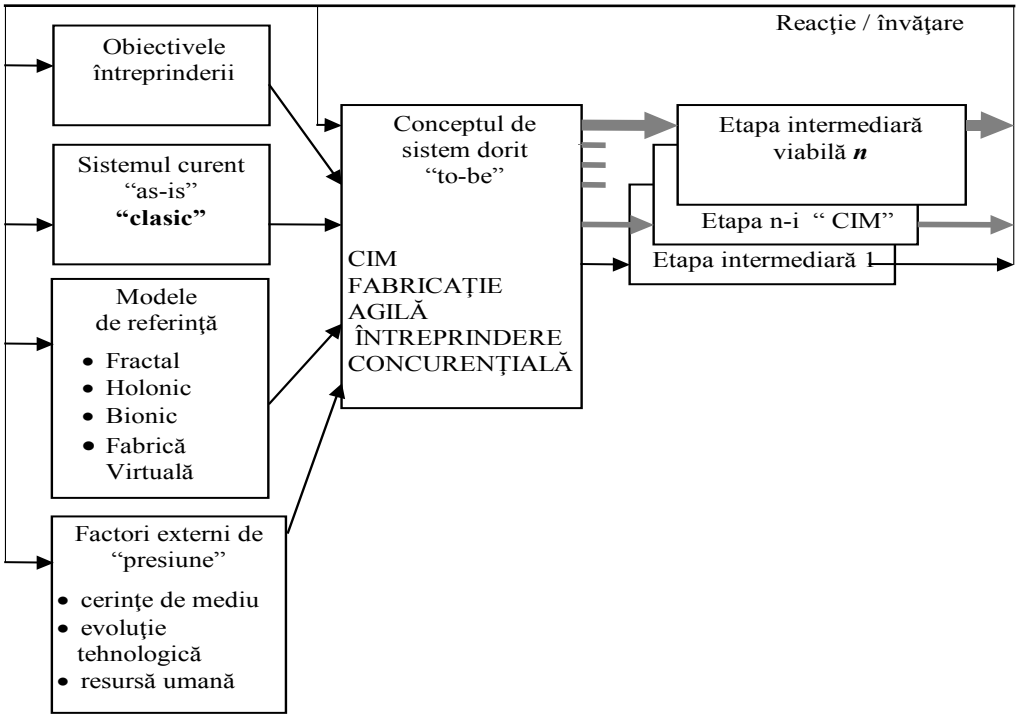


Fig.1.3. Sistemul de fabricaţie ca sistem cu învăţare

Evoluţia sistemului de fabricaţie de la o stare la alta se poate constitui într-o succesiune de schimbări pas–cu–pas, după fiecare schimbare urmând o etapă de evaluare a rezultatelor şi a efortului depus.

Pentru obţinerea schimbărilor necesare la nivel organizatoric, proiectarea şi reproiectarea sistemului de fabricaţie impune stabilirea unui grup de planificare, care devine elementul central în proiectarea şi implementarea sistemului integrat de fabricaţie.

Aşa cum s-a precizat, o întreprindere de producţie industrială discretă este o organizaţie complexă care depinde de–şi interacţionează cu–un număr mare de alte organizaţii. În scopul de a utiliza efectiv conceptele de sistem deschis este adesea util să se dezvolte modele simple, intuitive care descriu elementele subsistemului precum şi relaţiile dintre ele.

Această metodă de abordare a devenit comună pentru abordarea şi înţelegerea pe baze euristice a sistemelor complexe. O înţelegere euristică a sistemului se asociază cu o percepţie preliminară a naturii sistemului care poate fi utilizată ca punct de pornire în stimularea investigaţiilor referitoare la sistem. Altfel spus, o înţelegere euristică a sistemului „stabileşte etape" pentru explorarea sistemului şi pentru descoperirea de noi informaţii referitoare la el.

Din experienţă, calea pe care o urmează procesul de rezolvare a unei probleme este ***primitiv - complicat - simplu***.
Într-o asociere cu sistemele de fabricaţie:

- ***primitiv*** poate fi soluţia clasică (taylor-istă);
- ***complicat*** – soluţia CIM;
- ***simplu*****,** neavând o denumire, i se poate spune ***întreprinderea virtuală ca distribuţie geografică şi agilă în organizare***.

In figura 1.4. se propune o schema funcţională pentru sistemul de fabricaţie integrat în mediu, conţinând următoarele module :

- procesare informaţii;
- modelare fabrică;
- analiză strategică – raţionament cognitiv;
- adoptare decizii;
- producţie;
- interactionează permanent cu baza de date şi cunoştiinţe ale întreprinderii.

Modelul fabricii (lumii) este plasat central. Sistemul poate repeta continuu ciclul *sesizează-decide-acţionează* sub diferite orizonturi de timp.
Subsistemul de procesare informaţii asigură achiziţia datelor specifice integrării întreprinderii în mediul de afaceri şi adaptarea ei la macro-contextul social şi tehnologic.

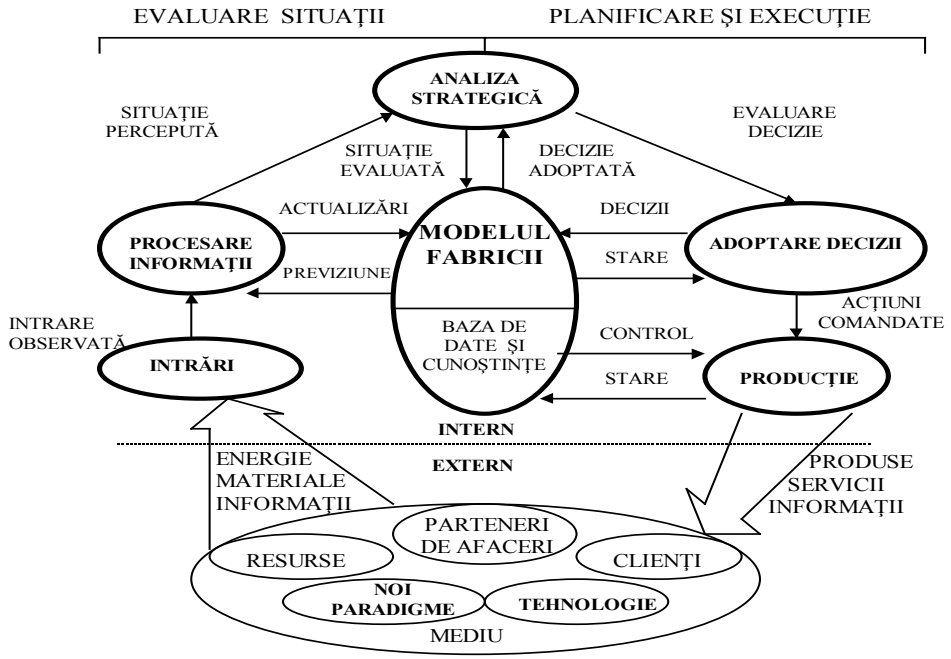


Fig. 1.4. Schema funcţională a
sistemului de fabricaţie integrat în mediul de afaceri

Funcţia de modelare a fabricaţiei

- păstrează modele care descriu mediul extern, diverse clase de modele de întreprindere, modelul curent de întreprindere ("as-is"), istoria evoluţiei întreprinderii ca macro-entitate şi istoria stării interne a sistemului întreprindere;
- actualizează modelul lumii;
- sprijină procesul de previziune al sistemului de fabricaţie.

Funcţia de adoptare a deciziei este cea care se ocupă de realizarea planurilor şi de îndeplinirea lor prin controlul asupra subsistemului de producţie şi asupra evoluţiei sistemului de fabricaţie, în situaţia în care de fapt se face referire la procesul de proiectare a unui sistem de fabricaţie. *Funcţia de analiză strategică* este, de fapt, o funcţie de judecată a valorii care evaluează atât starea observată a lumii cât şi previziunile făcute. Sunt calculate costuri, riscuri, şi beneficii atât pentru situaţii observate cât şi pentru activităţi planificate. Această funcţie asigură baza pentru alegerea unei acţiuni în favoarea alteia, utilizarea unui obiect în favoarea altuia, schimbarea modelului de referinţă, adoptarea de noi strategii.

1.2.2. MODELUL CLASIC DE ORGANIZARE A FABRICII

Structura ierarhică clasică a unei companii se prezintă în figura 1.5.

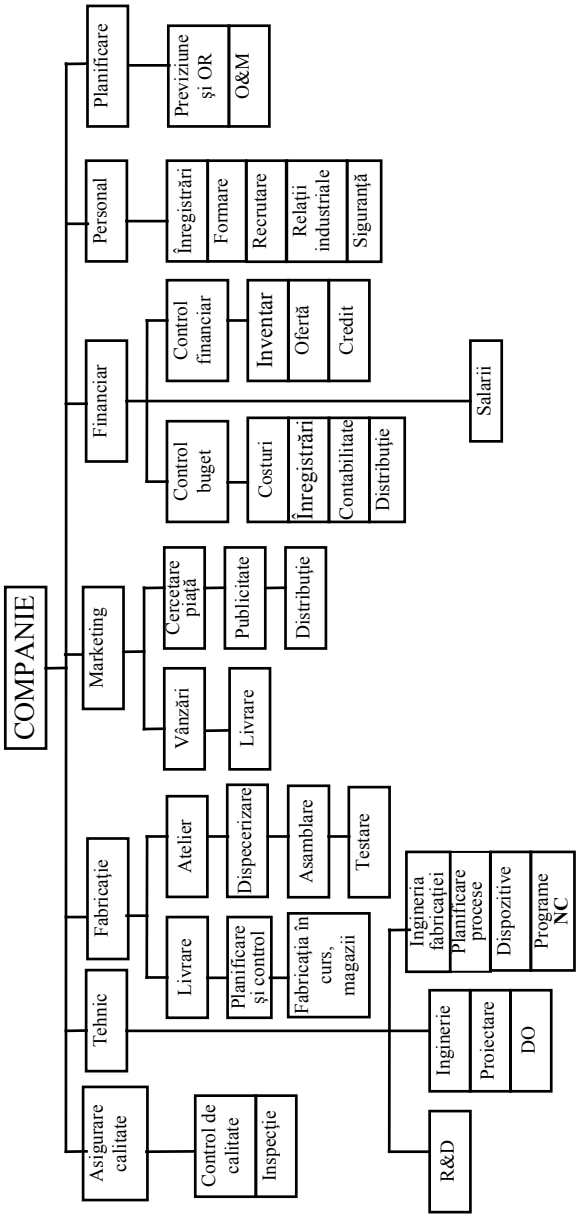


Fig.1.5. Modelul clasic de întreprindere industrială

Dintre elementele esenţiale ale modelului clasic de întreprindere pot fi considerate [Kidd,88;Gibson,92]:

- *reducţionismul* – sistemul este descompus în părţi componente care sunt tratate separat;
- *concepţia asupra optimizării componentelor* – se consideră că optimizând părţile se optimizează sistemul;
- *absolutizarea căii de succes* – se consideră că există o singură cale de a rezolva / a face bine un lucru, iar ea poate fi determinată prin analiză;
- *rigiditatea în definirea obiectivelor* – problemele şi obiectivele se dau; sarcina este de a găsi soluţii;
- *tratarea mecanicistă a resursei umane*; oamenii sunt trataţi ca maşini complexe şi atât, motivarea resursei umane este pur financiară, iar tehnologia nu este dezvoltată cu referire la resursa umană;
- *organizarea centralizat-ierarhică* - decizia se ia central, angajatul trebuie să execute ordinul de producţie, angajatul nu e implicat în procesul de adoptare a deciziei;
- *concepţia asupra execuţiei (aceasta nu necesită gândire);* prin analiză se poate face tot ceea ce trebuie făcut; la limită se consideră că nu sunt necesare experienţa, cunoştinţele tehnologice, intuiţia etc.;
- *ruperea individului de contextul social (recompensa individuală, pentru efort individual);* oamenii sunt recompensaţi pentru ceea ce fac individual, nefiind trataţi ca făcând parte dintr-o echipă.

Modelul de preţ de cost asociat acestui tip de organizare este cel al *„economiei la scară”,* care se bazează pe cicluri succesive în care standardizarea şi automatizarea fabricaţiei conduce la micşorarea costurilor unitare şi la scăderea preţurilor, ceea ce determină extinderea pieţei de desfacere care, la rândul ei face să crească cererea, şi ciclul se repetă (figura1.6.).

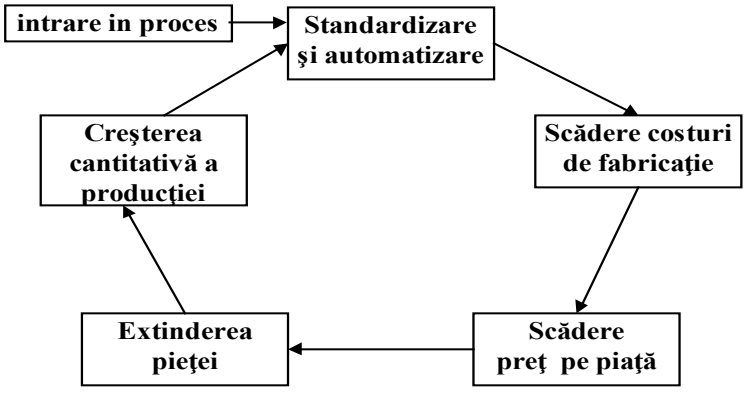


Fig.1.6. Modelul economiei de scară

1.3. INTEGRAREA FABRICAŢIEI PRIN CALCULATOR

Fabricaţia Integrată prin Calculator - CIM (Computer Integrated Manufacturing) este asocierea dintre *tehnologia informaţiei* şi *activitaţile de fabricaţie* având ca obiectiv creşterea productivităţii prin gestiunea eficientă a informaţiei şi a vitezei de reacţie a întreprinderii industriale faţă de cererile pieţii, relativ la dimensiunile funcţional, informaţional şi organizaţional care sunt părţi ale unui întreg total denumit întreprindere industrială.

Orientarea tehnologică a avut un rol deosebit de important în fabricaţie, cu un maxim în perioada anilor ’80.

Este esenţial a se face diferenţierea dintre tehnologiile bazate pe computer şi integrarea fabricaţiei cu computerul.

Aspectul esenţial al CIM-ului este acela de *integrare cu calculatorul, pe platforme funcţionale, a tuturor activităţilor* dintr-o companie (marketing şi vânzări, proiectare, planificare producţie, fabricaţie, cercetare-dezvoltare, management financiar). Trebuie subliniat că, în cadrul impus de conceptul tradiţional de fabricaţie, CIM-ul s-a focalizat asupra rezolvării unor probleme tehnice : sisteme şi protocoale de comunicaţii, baze de date, dezvoltarea unor interfeţe standard pentru date, sisteme de control bazate pe computere, etc.

Din punct de vedere al utilizării datelor, CIM-ul este o soluţie tehnică complexă bazată pe utilizarea computerului, a programelor specializate şi a sistemelor de comunicaţii, care permit memorarea datelor, întreţinerea acestora, asigură informaţiile curente necesare şi schimbul rapid de date între departamente

Din punct de vedere al impactului tehnologiei în automatizare, CIM-ul propune automatizarea proiectării, a planificării producţiei, fabricaţiei, depozitării şi oferă soluţii pentru controlul şi supervizarea centralizată, în timp real, a tuturor proceselor operaţionale.

Variabilitatea este o proprietate inerentă a sistemelor fizice şi nu o imperfecţiune reparabilă. Sistemele de control au rolul de a menţine variabilitatea între nişte limite acceptabile. Uzura şi defectarea echipamentelor, erorile datorate factorului uman,etc., conduc la disfuncţii ale resurselor şi la defecte de produs. Detectarea, diagnosticarea şi supervizarea sistemului de producţie pot conduce la schimbări, cel puţin pentru intervale limitate de timp, în procesul de fabricaţie.

Managementul proceselor combină activitatea de *monitorizare* (urmărire a variaţiilor, detectarea disfuncţiilor şi a defectelor), *diagnoza* (explicarea cauzelor care au condus la defecte), *controlul* (acţiuni menite a păstra variabilitatea sistemului în limitele prestabilite), *repararea* (acţiuni care conduc la înlăturarea disfuncţiilor şi a defecţiunilor) pentru optimizarea şi îmbunataţirea continuă a fabricaţiei. Managementul proceselor stabileşte cum se initiază, se desfasoară şi este monitorizat procesul de fabricaţie.

Managementul resurselor asigură planificarea şi ordonantarea proceselor, determină care operaţii de fabricaţie se desfasoară, când şi cu ce resurse.

Sistemele CIM dezvoltate în perioada anilor ‘80 integrează managementul de procese şi aplicaţii CAD cu module software care permit accesul la baza de date comună. Modelele de baze de date partiţionate şi accesul distribuit la date au fost elemente cheie, care au permis crearea de platforme de sistem distribuit şi ierarhizat, specifice CIM. În aceste platforme, un modul de aplicaţie îndeplineşte un set de proceduri, orientate pe funcţii care sunt generic din domeniul analizei activităţilor. Sunt utilizate modele care descriu cum sunt structurate entităţile informaţionale şi cum este asigurată legătura între ele.

Datele sunt integrate şi sunt accesibile oricărei aplicaţii din sistem. Integrarea parţială a aplicaţiilor a condus la inflexibilitate, costuri mari ale schimbării şi, nu în ultimul rând, la sisteme calitativ modeste.

Argumentul major este acela că împachetarea datelor în jetoane devine un obstacol important pentru un sistem de fabricaţie dinamic. Adăugarea de noi aplicaţii sau modificarea celor deja existente se constituie într-un proces laborios, iar baza de date centrală necesită operaţii de ajustare pentru a îngloba noile modele, rafinate şi extinse. Modificările induc modificări în alte aplicaţii, ceea ce conduce la creşterea complexităţii sistemului, la creşterea costului şi la probleme de calitate.

1.3.1. PLATFORME FUNCŢIONALE PENTRU INTEGRARE ACTIVITĂŢI

Activităţile dintr-o companie, de la fabricaţia propriu-zisă la previziune şi managementul central („top management”) pot fi reprezentate prin piramida organizării. Fiecare activitate utilizează date proprii şi necesită acces la datele altor activităţi.

O clasificare posibilă a tipului de date utilizate este :

- *date funcţionale*; utilizate de grupurile de specialişti ca date, informaţii şi cunoştinţe proprii, ce pot fi considerate ca făcând parte din baza de cunoştinţe a întreprinderii;
- *date produs*; generate de grupurile de specialişti (de exemplu: pentru departamentul de proiectare sunt datele despre produse şi subansamble);
- *date operaţionale;* necesare controlului operaţiilor unei fabrici (ex.: planuri, instructiuni etc.);
- *date de performanţă*; pentru monitorizare şi control cu reacţie.

În figura 1.7. se observă o multitudine de căi de transmitere a datelor (organizate ca date de proiectare, date de fabricaţie, date de producţie) între diverse activităţi. Departamentul de achiziţii necesită informaţii de la nivelul controlului producţiei, iar controlul producţiei utilizează datele venite de la nivelul

"atelier". Sensul acestor date este de la bază spre nivelurile ierarhice superioare. Pentru control, sensul datelor este de la vârful piramidei spre bază. Informaţia este elementul central al unei întreprinderi de fabricaţie, iar utilizarea datelor (memorarea, accesul, actualizarea etc) subliniază nevoia esenţială ca *întreprinderea să se constituie într-o structură integrată* având rolul principal de a face informaţia disponibilă când este nevoie, unde este nevoie şi în formatul cerut, accesibile inter-aplicaţii.

Platformele funcţionale pentru integrare activităţi, denumite generic CAx (ComputerAided; x = D (Design); x = M (Manufacturing);
x = O (Organization); x = P (Planning) etc) au fost dezvoltate izolat unele faţă de altele, cu implicaţii majore în modul de reprezentare a datelor şi în schimbul de informaţii, conducând la soluţii „scumpe", la dezvoltarea de interfeţe specializăte, la împachetarea datelor în jetoane, capabile să asigure interfaţarea între aceste platforme.

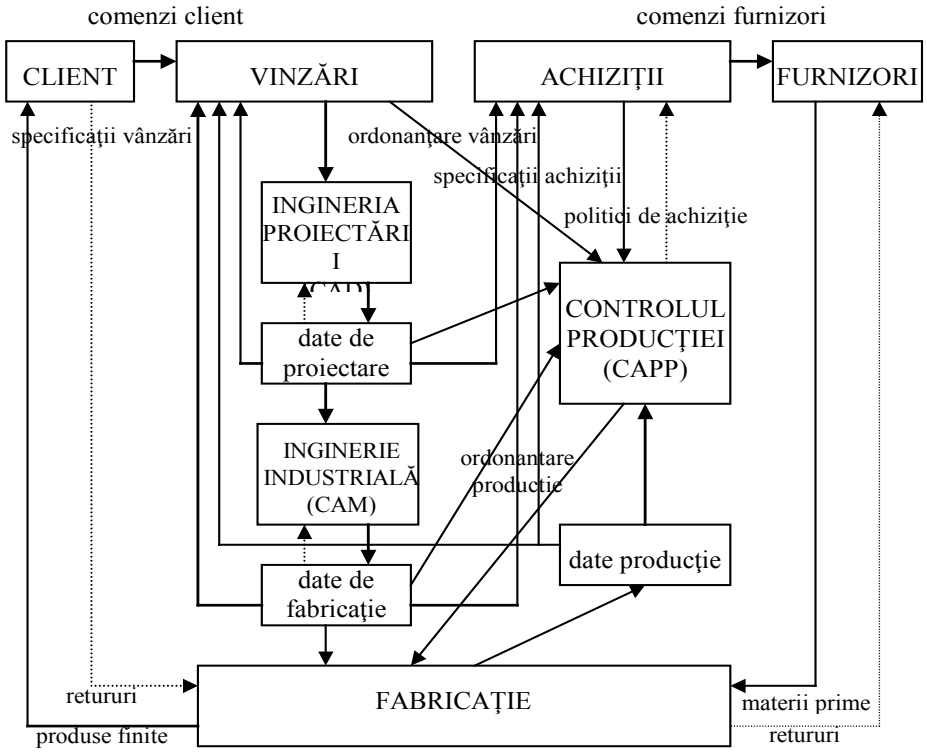


Fig. 1.7. Fluxuri de date

Nevoia de integrare a condus la crearea de *platforme de execuţie pentru integrarea aplicaţiilor de fabricaţie* (de exemplu: CIMECE - *Computer integrated Manufacturing & Engineering Computing Environment*). Proiectarea şi implementarea acestor platforme ţin cont de principalele caracteristici ale aplicaţiilor CIM:

- *aplicaţii de tip loturi*; odată pornite, necesită un schimb minim de informaţii, iar la încheiere generează rezultate disponibile în sistem;
- *aplicaţii on-line*; puternic interactive, care rămân permanent active comutând între starea de aşteptare şi cea efectivă de lucru;
- *activităţi on-line "orientate eveniment";* deciziile în procesul de fabricaţie sunt luate pas cu pas pe baza evenimentelor apărute, ceea ce asigură un răspuns rapid la schimbările lumii externe;
- *solicită date consistente* pentru reprezentarea stării sistemului în scopul controlului;
- trebuie să dispună de *facilităţi de integrare prin comunicaţii* a unei game largi de dispozitive şi echipamente;
- trebuie să asigure *siguranţa operării* la nivel de aplicaţie ţintă;
- *sunt capabile să utilizeze o gamă variată de informaţii* .

Cauzele obiective care determină complexitatea acestor platforme de integrare activităţi pot fi sintetizate astfel:

- CIM-ul este un sistem compus din multiple aplicaţii puternic interdependente, care manipulează şi utilizează o cantitate mare de date;
- CIM-ul este structurat adesea ierarhic, fiecare nivel din această ierarhie introducând cerinţe şi constrângeri;
- multitudinea de echipamente, computere, controlere, automate programabile, CNC etc. care sunt necesare aplicaţiilor diverse din CIM, generează necesitatea de a crea un număr mare de interfeţe specializate şi de a integra un mare număr de standarde de interconectare şi transfer de informaţii pentru control şi stare;
- aplicaţiile sunt complexe, interfeţele operator necesită costuri mari de dezvoltare, sunt dificil de întreţinut, iar instruirea operatorilor este dificilă;
- dezvoltarea aplicaţiilor necesită costuri mari fiind în general aplicaţii dedicate;
- procedurile de tratare a erorilor, refacere sistem şi management date sunt complexe şi adesea inconsistente.

Cerinţe pentru o platformă de integrare activităţi :

- să asigure ca accesul la resurse (accesul datelor, comunicaţiile) să se facă in mod *transparent* (fără a fi necesară cunoaşterea detaliilor de nivel scăzut);

- să fie *deschisă,* ceea ce presupune ca specificaţiile pentru servicii şi interfeţele trebuie să fie accesibile şi să permită dezvoltarea produsului de către terţi;
- să poată rula ca *aplicaţii distribuite* pe mai multe calculatoare, ţinând cont că majoritatea aplicaţiilor industriale sunt aplicaţii distribuite;
- sa fie *modulara*; blocurile functionale componente pot fi instalate in functie de cerintele aplicatiei;
- să fie *extensibilă;* platforma să permită adaugarea de noi blocuri funcţionale;
- să permită *configurarea dinamică* a aplicaţiilor (chiar şi in timp ce ele rulează);
- să permită *rularea/integrarea* aplicaţiilor deja existente, prin modificări minime (să raspundă specificaţiilor de sistem deschis);
- să aibă mecanisme care să asigure *consistenţa datelor şi siguranţa transmiterii* lor;
- să aibă mecanisme de *refacere* stare după căderile de sistem;
- să *optimizeze viteza de rulare a aplicaţiilor*, care să le facă compatibile cu cerinţele de timp real ale lumii externe;
- să aibă *mecanisme de securitate date* faţă de accese neautorizate;
- să aibă *interfaţa prietenoasă* cu operatorul şi să ofere facilităţi pentru crearea /modificarea acestora.

Figura 1.8. prezintă o organizare a activităţilor dintr-o întreprindere sub forma unei *galaxii* cu legături prin fluxuri de date şi materiale, orientată pe platforme funcţionale.

1. CAE: ingineria asistată de calculator a produsului şi a fluxului tehnologic de realizare a acestuia.

Activităţi specifice :

1.1. *CAD : proiectarea asistată de calculator*
- Proiectare;
- calcule tehnice;
- modelări/simulări;
- desene;
- lista de materiale.

1.2. *R&D : cercetare / dezvoltare produs*
- investigare/testare/dezvoltare noi materiale,produse, procese tehnologice;
- proiectare / modelare / simulare şi analiză noi produse;
- realizare şi testare prototipuri;

1.3. *Dezvoltarea fabricaţiei*
- dezvoltare de noi metode de fabricaţie;
- adaptare tehnologii de fabricaţie;

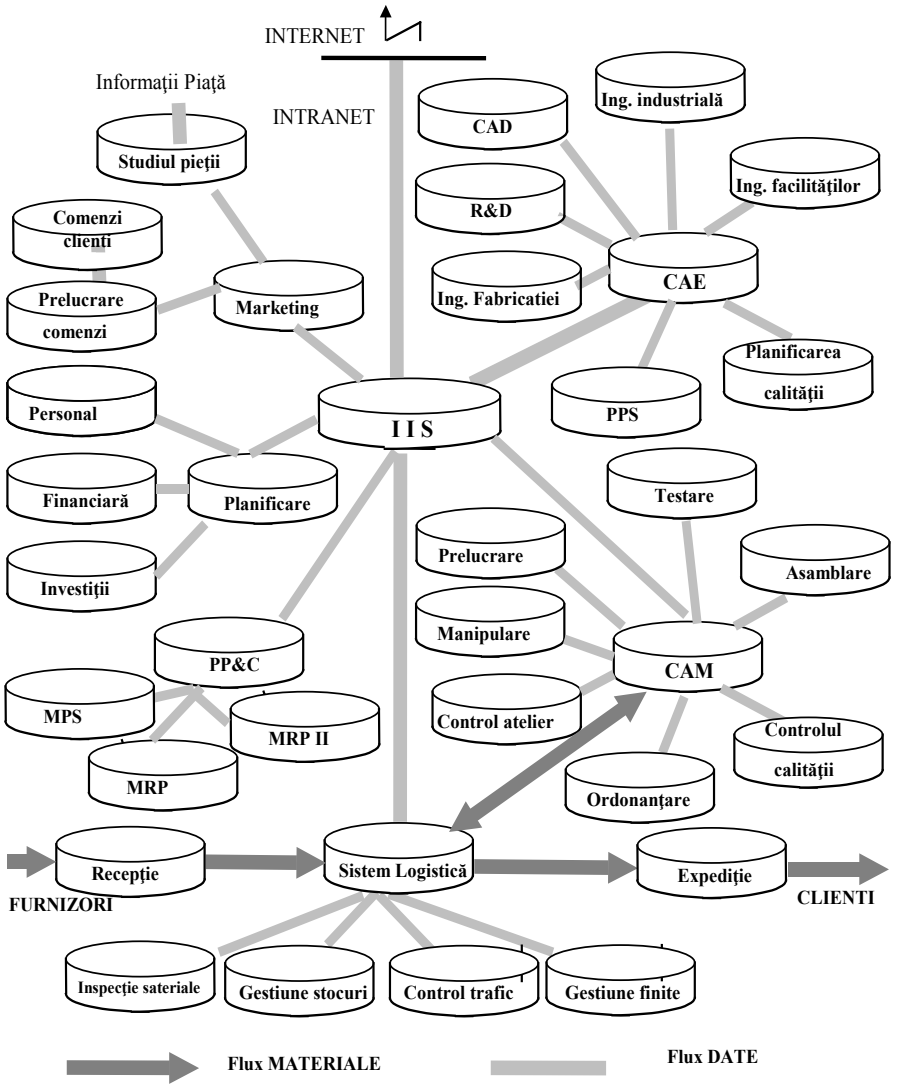


Fig.1.8. Model de galaxie CIM

1.4. *Ingineria fabricaţiei*

- *validează fabricabilitatea produsului* proiectat;
- *CAPP (PPS):* planificarea proceselor;
- *planificarea proceselor de prelucrare* (planificarea materiilor prime şi a materialelor a secvenţei de procese tehnologice, a sculelor şi dispozitivelor, a parametrilor de prelucrare);

- *planificarea proceselor de asamblare* (stabilirea relaţiilor de precedenţă, definirea planului de acţiune, alegerea echipamentelor necesare);
- *planificarea proceselor de testare* (stabilirea metodelor şi a secvenţelor de test, alegerea echipamentului necesar);
- *generarea planurilor* de prelucrare, asamblare, testare;
- selectare/proiectare scule, dispozitive de prindere;
- responsabilă de aplicarea noilor tehnologii de fabricaţie implementate de ingineria facilităţilor;
- proiectarea sarcinilor, analize de utilizare timp şi mobilitate personal, proiectare şi analiză a operaţiilor de fabricaţie, studii ergonomice etc., care uzual ţin de ingineria industrială.

1.5. *Ingineria facilităţilor*

- planificarea instalării noilor echipamente;
- planificarea fluxului de materiale şi a echipamentelor auxiliare;
- planificarea spaţiilor de depozitare;
- are implicaţii şi la nivelul fabricaţiei efective având atribuţii de planificare consum energetic, resurse auxiliare tehnologice (aer comprimat, abur, caldură, lumină etc.) şi de precizare a noilor disponibilităţi de echipamente.

2. **PP&C : Planificarea şi Controlul Producţiei**

Activităţi specifice :

2.1. ***MPP : Planificarea (si ordonanţarea) Principală a Producţiei***

- planifică programul de fabricaţie pe bază clienţilor existenţi, a previziunilor legate de piaţă şi a deciziilor de afaceri;
- dimensionează loturile;
- generează planul de producţie orientat timp-faze (ordonanţare producţie) care este transmis modulului de planificare a resurselor materiale(MRP).

2.2. *MRP* ***: Planificarea (şi ordonanţarea) Resurselor Materiale***

MRP II : Planificarea (şi ordonanţarea) Resurselor de fabricaţie

- asigură detaliile despre fiecare produs ce urmează a fi realizat denumită *"explozia în componente*";
- planifică procurarea materialelor necesare (pe baza informaţiilor referitoare la stocurile curente şi la furnizorii potenţiali);
- asigură planificarea şi controlul inventarului;
- dimensionează economic mărimea loturilor (secvenţierea producţiei);
- planifică emiterea ordinelor către furnizori;
- planifică recepţionarea (primirea şi controlul) componentelor;
- monitorizează starea capacităţilor de producţie şi planifică incărcarea acestora;
- stabileşte maşinile de rezervă şi rutele alternative;
- asigură ordonanţarea şi re-ordonanţarea la nivel de ateliere.

2.3. *Aprovizionare*

- selectare furnizori;
- generare ordine de aprovizionare.

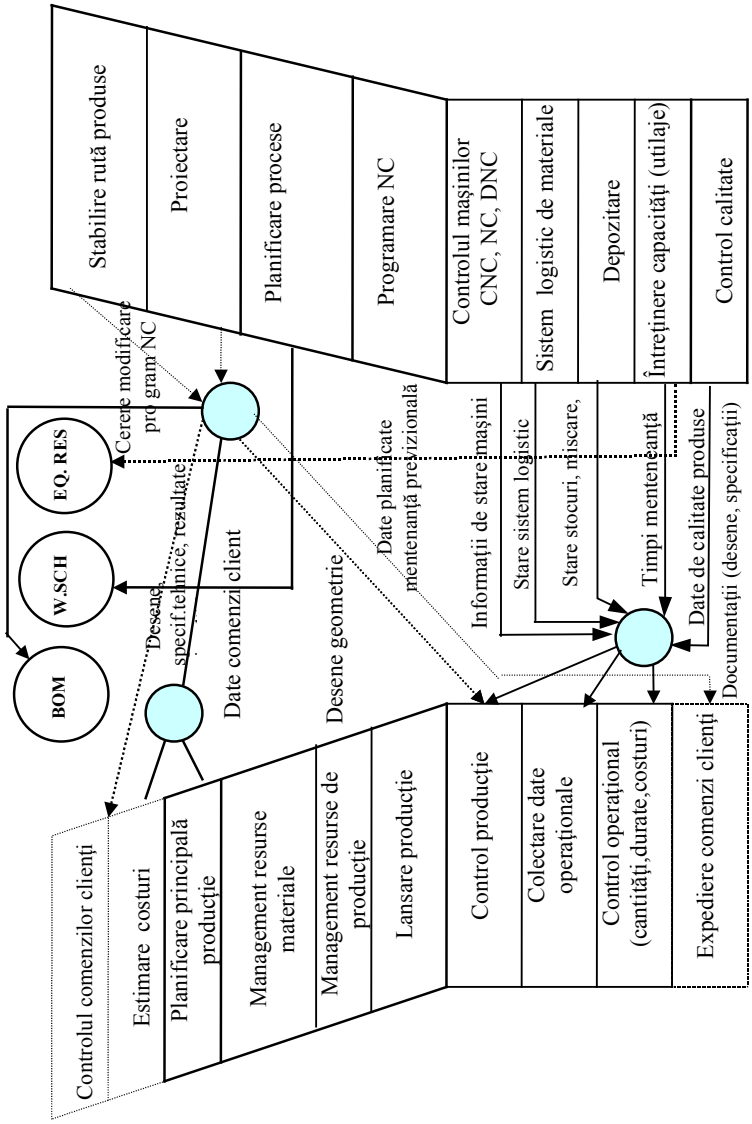


Fig. 1.9. Flux informaţii CAD / CAM / PP&C

2.4. *Controlul producţiei*
- emite planul de ordonanţare a activităţilor către ateliere şi controlează îndeplinirea lui;
- asigură controlul operaţional (cantităţi, durate, costuri).

3. **CAM : Fabricaţia asistată de calculator**
Activităţi specifice :

3.1. *Planificarea tehnologică şi detalierea metodelor de fabricaţie, din atelier:*
- planificarea secvenţelor de operaţii ("ruta" produsului);
- planificarea în detaliu a operaţiilor individuale (procese).

3.2. *Ordonanţarea producţiei şi controlul fabricării produsului*
- ordonanţare şi re-ordonanţare la nivel de atelier;
- controlul procesării ordinelor de fabricaţie la nivel de atelier;
- achiziţie date fabricaţie;
- evaluare date fabricaţie;
- control flux materiale;
- control transport materiale;
- control înmagazinare;
- control magazii intermediare.

3.3. *Automatizarea şi controlul cu calculatorul a echipamentelor de fabricaţie*
- maşini CN/CNC;
- dispozitive de transport şi alimentare automată;
- roboţi şi vehicule autonome (AGV);
- sisteme integrate de producţie (FMS).

1.3.2. ARHITECTURI DE REFERINŢĂ CIM

O arhitectură, în sensul proiectării, este un vehicul pentru exprimarea conceptuală a formei, regulilor, legăturilor, direcţiei strategice şi scopului, care exclude datele nesemnificative care pot împiedica conciziunea. Ca strategie, o arhitectură este un indicator direcţional consistent, al cărui scop nu este acela de *a rezolva o problemă* ci de a structura o situaţie, astfel încât problemele apărute să fie solvabile [Minzberg,87]. Arhitectura de referinţă a unei întreprinderi modelează istoria întregului ciclu de viaţă a integrării întreprinderii, de la concepţia iniţială a antreprenorului, la definirea ei, specificarea şi proiectarea funcţională, proiectarea în detaliu, implementarea fizică şi operarea. Arhitectura este *elementul integrator* al tuturor aspectelor legate de proiect.

Arhitectura are rolul de a preciza:
- modalitatea de adoptare a deciziei;
- procesele de afaceri;

- activităţile întreprinderii;
- fluxul de informaţii;
- fluxurile de materiale şi energie.

Dintre beneficiile utilizării unei arhitecturi de referinţă se pot preciza :

- posibilitatea de a verifica consistenţa tuturor funcţiilor descrise şi a obiectelor (procese de afaceri, date, materiale, resurse), la toate nivelurile de detaliu;
- posibilitatea simulării modelelor utilizate pentru întreprindere, la fiecare nivel de detaliu;
- posibilitatea schimbării modelului atunci când se modifică procesele de afaceri, metodele sau dispozitivele;
- posibilitatea utilizării modelelor pentru iniţializare, monitorizare şi control al execuţiei operaţiilor etc.;
- suport pentru alocarea şi realocarea resurselor pe durara procesului de afaceri;
- suport pentru generarea modelului curent de întreprindere.

Activităţile realizate de-a lungul tuturor meta-fazelor ciclului de viaţă al întreprinderii sunt, în general, rezultatul unor proceduri de moment, utilizate de proiectanţi şi experţi pe măsura dezvoltării sistemului integrat. Calitatea proiectării depinde esenţial de experienţa persoanelor implicate, situaţie accentuată de inexistenţa unui formalism puternic care să susţină aceste activităţi. Rezultatul este: *soluţii neorientate afaceri* şi *soluţii prea costisitoare* în sensul că aplicarea rezultatelor obţinute (procedurile stabilite) nu conduce la rezultate având performanţe apropiate de cele care au validat procedurile. Nevoia de imbunătăţire a formalismului utilizat în proiectarea întreprinderii subliniază necesitatea capturării cunoştinţelor esenţiale pentru această activitate.

Eforturile de formalizare a proiectării deciziilor, asociat cu activităţile specifice fiecărei faze a ciclului de viaţă a întreprinderii integrate, susţinute de evoluţia tehnologiilor informatice s-au concentrat în crearea unor arhitecturi de referinţă.

Arhitectura de referinţă se constituie în mijloacele necesare, prin care activităţile din ciclul de viaţă pot fi structurate, astfel încât prin manipularea modelelor propuse să se asigure *generarea* întreprinderii integrate de fabricaţie [Aguiar,95].

Modelele reflectă demersul de a captura maximum de cunoştinţe formale, care să garanteze calitatea rezultatului proiectării deciziilor.

Arhitectura de referinţă defineşte *regulile* şi *structurile* care sunt utilizate în ciclul de viaţă al întreprinderii integrate de prelucrare (IIP); ea cuprinde aplicarea unei metodologii (reguli) într-un context dat (structură), cu scopul de a obţine un

model de referinţă, prototip (*pattern* de descriere) a unui anumit tip de organizare al întreprinderii.

Arhitectura de referinţă presupune *aplicarea unei metodologii* (a unor reguli), într-un *cadru de modelare* (structură) dat în scopul obţinerii unui model de referinţă a unui anumit tip de întreprindere. Scopul esenţial al unei arhitecturi de referinţă este acela de a permite generarea unei soluţii particulare de întreprindere, *arhitectură particulară,* din cerinţele de afaceri.

Arhitectura se poate defini ca o descriere (model) al structurii unui obiect sau entităţi fizice sau conceptuale.
Prin prisma acestei definiţii se pot stabili două tipuri de arhitecturi, care au ca scop integrarea entităţilor fabricaţiei sau întreprinderii şi care orientează:

- *proiectarea dispunerii structurale a sistemului fizic*, cum ar fi sistemele computerizate de calcul, ca parte a unui sistem complet integrat de întreprindere; din această clasă fac parte şi *arhitecturile ierarhice de control;*
- *organizarea dispunerii structurale a dezvoltării şi implementării unui proiect sau program de tipul întreprindere integrată***,** fabricaţie integrată sau alt program de dezvoltare; generic această clasă va fi denumită *arhitectură de referinţă.*

1.3.3. CICLUL DE VIAŢĂ AL ÎNTREPRINDERII CIM

Fiecărui nivel din organizarea ierarhică a întreprinderii îi este asociată o faza din ciclul de viaţă. Se consideră că toate cunoştinţele necesare funcţionării întreprinderii sunt cuprinse în modelele asociate fiecărui nivel. Proiectarea unui sistem CIM este un proces iterativ, care trebuie să structureze procesul de adoptare al deciziei în faze diferite ale ciclului de viaţă al întreprinderii. La nivelul fiecărei faze se impun cerinţe de proiectare, iar rezultatele unei faze asigură intrări pentru faza următoare în ierarhie. Ciclul de viaţă al unui sistem CIM se consideră ca având patru meta-faze, puternic corelate cu fazele definite în [Heim,94]. Sensul spiralei este acela de scădere a riscului, odată cu parcurgerea ciclului de mai multe ori şi stabilizarea sistemului.

Planificarea Strategică:

- domeniul managementului de vârf (obiective, strategie, performanţe întreprindere);
- asigură analiza strategică în vederea creşterii performanţei globale a întreprinderii;
- utilizează modele de întreprindere care au grad mare de abstracţie şi se referă la modelarea şi simularea mediului de afaceri;

- utilizează modele ale lumii externe întreprinderii;
- rezultatele de la acest nivel dau răspunsul la întrebările: *când* se investeşte, în *ce constă* investiţia, *unde* se face investiţia, *cât costă efortul* şi care sunt *rezultatele* estimate.

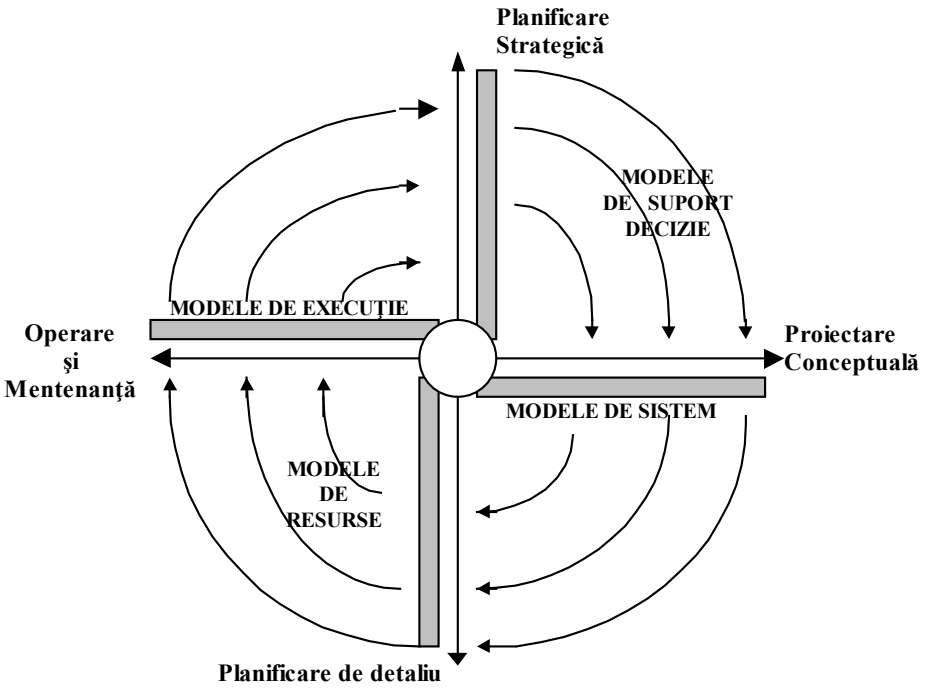


Fig. 1.10. Ciclul de viaţă al sistemelor CIM

Proiectarea Conceptuală:
- domeniul analiştilor de sistem;
- se studiază în detaliu domeniul selectat de planificarea strategică;
- utilizează modele formale de sistem;
- rezultatele:
  - definirea tipului de soluţie (ce se integrează şi cum);
  - soluţia descrisă în termeni de cerinţe funcţionale.

Proiectarea de detaliu şi Implementarea:
- domeniul proiectanţilor de sistem;
- se specifică soluţiile fizice, iar pe baza lor se defineşte sistemul ce urmează a fi implementat; stabileşte tipul de sistem fizic implicat (decizia se face / se cumpară);

- rezultatele : set de modele care stabilesc cum operează întreprinderea (modele executabile).

Operare şi Intreţinere
- domeniul infrastructurii de întreprindere, care sprijină introducerea modelelor şi crează cadrul pentru execuţia lor;
- asigură managementul activităţilor operaţionale într-o întreprindere;
- rezultate: acţiuni realizate de componentele operaţionale care utilizează serviciile de infrastructură.

Concluzii:
- necesitatea asigurării *consistenţei între modelele* utilizare la diverse niveluri;
- necesitatea asigurării *legăturilor formale dintre modele* prin suprapunerile dintre acestea sau prin formate comune intrare/ieşire, asigurate direct sau prin postprocesare; interfeţele dintre modele crează o permanentă problemă datorită lipsei unor standarde în domeniu;
- necesitatea asigurării *integrării şi interoperabilitaţii diverselor modele.*

1.3.4. PLANIFICARE ŞI CONTROL IERARHIC

Într-un sistem ierarhic de planificare şi control se pot identifica un număr de niveluri, care diferă între ele prin orizontul de timp, gradul de abstractizare, structurile de date utilizate, comunicaţia dintre modulele funcţionale aflate pe acelaşi nivel ierarhic, viteza de actualizare a datelor utilizate de către fiecare nivel. Generic, arhitectura ierarhică de planificare şi control are în componenţă trei sau patru nivele [Rufels,93]:
- ***Planificarea strategică*** – Scopul planificării este acela de a asigura competitivitatea pe piaţă a întreprinderii. Activităţile asociate sunt: cercetarea de marketing, planificarea pe termen lung, previziunile legate de nevoia de investiţie de capital, planificarea facilităţilor;
- ***Planificarea operaţională*** – Este focalizată pe activităţile de planificare tehnologică şi organizatorică, gestiunea comenzilor clienţilor, ingineria proiectării, planificarea proceselor de fabricaţie, marketing, ordonanţarea şi controlul ordinelor de producţie, distribuţia;
- ***Controlul fabricaţiei*** – Acoperă funcţiunile de management stocuri, logistică de fabricaţie, fabricaţie propriu-zisă, asamblare, control de calitate;
- ***Controlul la nivel de maşină*** – Este controlul la nivelul echipamentelor de fabricaţie.

Descompunerea ierarhică oferă avantajul unei structuri organizatorice consistente care sprijină noţiunile de decompoziţie şi abstractizare. Ea se referă la o cale naturală de a conduce sistemele complexe de fabricaţie [McLean,83]. Esenţa arhitecturii ierarhice de control este modul în care sunt aranjate controlerele într-o ierarhie. Adesea, structura de comandă este o ierarhie de tip arborescent, în care fiecare controler are un superior şi nici unul sau mai mulţi subordonaţi. Un sistem ierarhic de control nu permite explicit schimbul de comenzi între controlerele aflate pe acelaşi nivel în ierarhie, dar permite acestora să utilizeze în comun aceleaşi informaţii prin multiple mecanisme de comunicaţie. Într-o arhitectură de control ierarhică, execuţia task-urilor este uzual rezultatul unei comenzi venite de la superior către subordonat corelată cu raportarea de jos în sus a completării task-ului. Coordonarea sau sincronizarea cu alte activităţi pot fi iniţiate prin comenzi.

O unitate aflată pe un nivel ierarhic intermediar are atribuţii referitoare la :

- recepţionarea informaţiilor de stare de la nivelul inferior;
- transmiterii comenzilor către nivelul inferior;
- transmiterea informaţiilor de stare la nivelul superior;
- indeplinirea comenzilor primite de la nivelul superior.

Unitatea aflată în vârful ierarhiei poartă responsabilitatea finală pentru îndeplinirea obiectivelor întregului sistem.

1.3.5. ARHITECTURĂ IERARHICĂ DE CONTROL

Modelul ierarhic de control NIST (*National Institute for Standards and Technology*) îşi propune crearea unor facilităţi de cercetare a sistemelor avansate de fabricaţie (AMRF), care să opereze independent de standardele legate de structurile hardware sau software disponibile la un moment dat pentru sistemele de control a fabricaţiei [Albus,81, Rufles,93]. Modelul propune un mediu dinamic de control al fabricaţiei şi proiectarea modulară. Planificarea şi controlul operaţiilor de fabricaţie este realizată prin intermediul unui sistem ierarhic de calculatoare şi a unui sistem senzorial. O mare parte a activităţilor, în special la nivelurilor de jos, sunt de timp real. Accesul la date este asigurat de către sistemul de infrastructură informatică, care înglobează în reţeaua de comunicaţii toate facilităţile de calcul din sistem. Se propune conceptul de „căsuţă poştală” (*mailbox*), ***locaţii virtuale de memorie***, pentru susţinerea schimbului de informaţii şi pentru simplificarea proiectării operaţiilor sistemului de control.

Modelul original adresează următoarele aspecte:

- ***managementul sistemului informatic***, care coordonează şi controlează întregul flux de date (control şi stare) din sistem;

- ***sistemul de proiectare şi planificare***, care pregăteşte documentele de fabricaţie;
- ***sistemul de control.***

Elementele esenţiale ale acestui model sunt structura ierarhică de planificare şi control, transparenţa şi partiţionarea făcută între management, planificare şi control.

Informaţiile despre comenzile de fabricaţie sunt introduse în sistem la nivelul de sus al ierarhiei, unde se stabilesc obiectivele globale şi strategia companiei. Comenzile iniţiază procesul de proiectare şi planificare a activităţilor. Procesarea comenzilor este rafinată pe măsura coborârii în ierarhie, astfel încât la baza ierarhiei se generează un set de instrucţiuni primitive de control, care operează direct echipamentele de fabricaţie. Procesul de generare a activităţilor specifice este descompus în trei componente [Albus,92]: ***alocarea job-ului(AJ), planificare (PL), execuţie (EX).***

O activitate este descompusă în subactivităţi de către un manager care stabileşte sarcini. Planificatorul fiecărui subordonat (PL1, PL2 şi PLi) comandă subactivităţile într-o secvenţă temporală. Fiecare subactivitate este efectuată de un executant (EX1, EX2 şi EXj). Subactivitatea comandată unui executant de pe un nivel ierarhic devine activitate, care la rândul ei, este descompusă în subactivităţi pentru nivelul inferior. Pe masură ce se coboară în ierarhie cerinţele de timp real se accentuează, ceea ce conduce la necesitatea precizării activităţilor principale ale sistemului de conducere: ***monitorizare*****a** stării sistemului de fabricaţie, ***diagnoza*** de sistem, ***re-ordonanţarea*** sistemului de fabricaţie.

Nivelul fabricii

La nivelul fabricii activităţile principale sunt marketingul şi stabilirea task-urilor de fabricaţie necesare pentru acceptarea unei comenzi. Estimarea costurilor şi planificarea capacităţii sunt elementele de bază pentru acceptarea unui ordin. Departamentele de vânzări, proiectare şi inginerie sunt văzute la nivel de fabrică datorită influenţelor pe care le au în acceptarea unei comenzi.

Nivelul de sistem

Activităţile specifice acestui nivel sunt cele de planificare procese, planificare producţie şi asigurare a calităţii. Deşi proiectarea în vederea fabricaţiei presupune integrarea proiectării activităţilor inginereşti cu planificarea de procese, funcţiile sunt plasate de către model, pe niveluri diferite. La acest nivel se stabileşte modul în care se va fabrica un produs - ce operaţii, în ce secvenţă, cu ce echipamente, dispozitive etc.

Nivelul celulei de fabricaţie

Acest nivel este considerat esenţial în cadrul modelului deoarece toate activităţile sunt orientate executarea planurilor de producţie. Ordonanţarea, pregătirea şi lansarea fabricaţiei sunt funcţiile esenţiale la acest nivel.

Nivelul echipamentelor

Reprezintă nivelul fizic de execuţie .

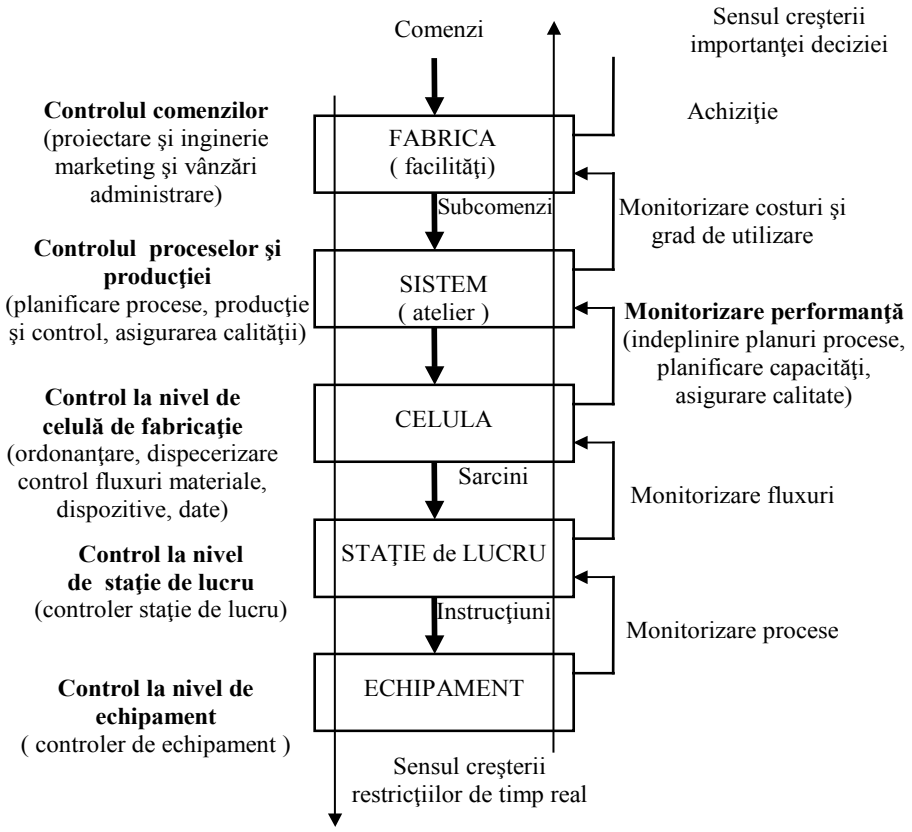


Fig. 1.11. Arhitectura ierarhică de control

Avantaje ale arhitecturilor ierarhice de control :

- modularitatea naturală a sistemului; fiecare controler poate fi tratat ca un modul; se facilitează dezvoltarea incrementală, iar soft-ul este mai uşor de elaborat şi de întreţinut;
- stabilitatea sistemului sintetizat;

- facilităţi pentru eliminarea disfuncţiilor prin întreruperea ierarhiei pentru zona afectată;
- permite lansarea cu frecvenţă diferită a controllerelor aflate pe diverse niveluri ierarhice, ţinând cont că frecvenţa cea mai mare o au acele controlere aflate pe nivelurile scăzute ale ierarhiei, acolo unde sunt impuse specificaţii de timp real.

Dezavantaje ale arhitecturilor ierarhice de control:

- efort mare de proiectare software;
- efort mare de modificare a ierarhiei;
- volumul mare de comunicaţii între controlere; informaţia trebuie să parcurgă întregul canal de comunicaţie într-o ierarhie;
- necesitatea coordonării interacţiunii între controlere;
- facilităţile pentru eliminarea disfuncţiilor prin întreruperea ierarhiei pentru o zona afectată, odată utilizate crează probleme prin deconectarea „bruscă" a întregului lanţ ierarhic aflat la niveluri inferioare modulului defect.

O clasă de arhitecturi ierarhice de control, de referinţă în literatura de specialitate, este arhitectura de sistem de control de timp real (**RCS**) şi specializări ale acesteia [Albus 81, 87, 91, 92] respectiv NASREM [Albus,89].

1.3.6. MODEL DE SISTEM DE CONTROL AL ACTIVITĂŢILOR DE PRODUCŢIE

Controlul Activităţilor de Producţie (**CAP**) poate fi definit ca un set de activităţi care asigură realizarea producţiei pe termen scurt şi controlul inventarului unei secţii, în concordanţă cu cerinţele impuse de sistemul de control al producţiei şi de gestiune a stocurilor la nivel de fabrică, ţinând cont de perturbaţiile la nivelul secţiei şi de influenţa mediului în care evoluează sistemul.

CAP este influenţat de doi parametri: complexitatea sistemului de fabricatie şi incertitudinile de mediu specific.

Modulul **CAP** se poate **instanţia** în fiecare nod al arhitecturii ierarhice de control, NIST, fiecare modul realizând următoarele funcţiuni:

- ***adaptarea*** – generarea şi actualizarea planurilor de producţie;
- ***optimizarea*** – evaluarea planurilor de producţie propuse şi oronanţarea sau
- re-ordonanţarea;
- ***dispecerizarea*** – transmiterea job-urilor către controlerele subordonate, monitorizarea comportării, diagnoza şi orientarea procesului de refacere.

Aceste funcţii definesc nivelurile unui controler, fiecare nivel operând la altă scară de timp şi utilizând alte tipuri de informaţii.

O arhitectură generică pentru **CAP**, la nivel de secţie conţine următoarele blocuri funcţionale :

- ***bloc de ordonanţare*** (***SCHEDULER***); realizaeaza planul de ordonantare a
- operaţiilor de producţie;
- ***bloc de transmitere*** (***DISPATCHER***); asigură coordonarea staţiilor de lucru, supervizează logistica de secţie, asigură realizarea planului de ordonanţare şi poate resecvenţia job-urile în cazul apariţiei unor disfuncţii;

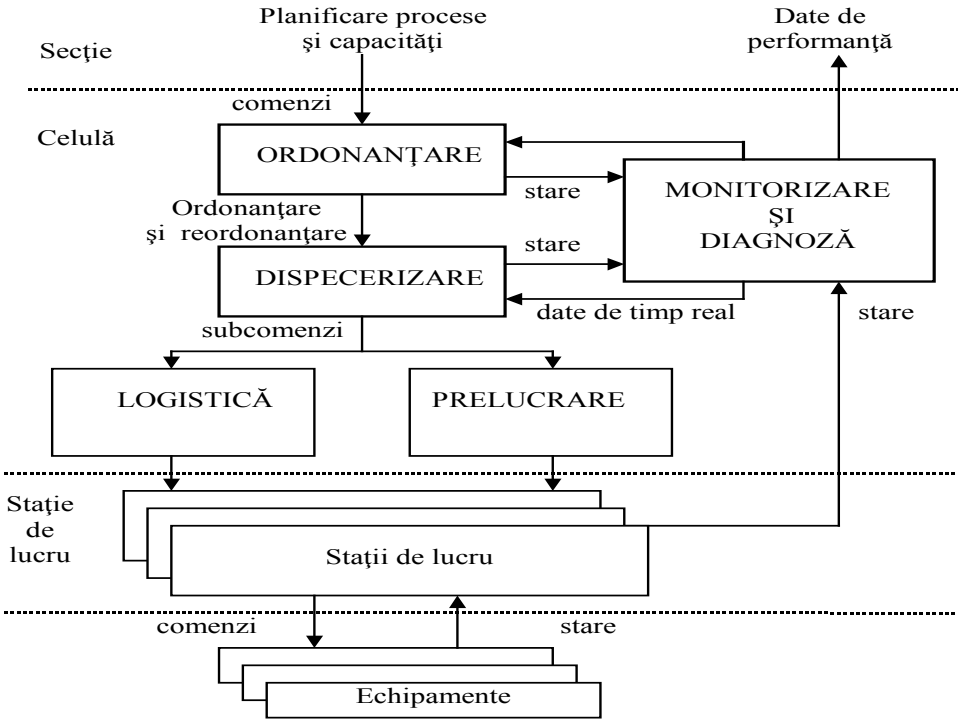


Fig. 1.11. Model de arhitectură generică pentru CAP

- ***bloc de monitorizare şi diagnoză*** (***MONITOR & DIAGNOSE***); asigură colectarea şi prelucrarea datelor de timp real de la nivelul secţiei şi conversia lor în formate utile procesului de adoptare decizie de la nivelul superior ierarhic precum şi diagnoza de sistem;
- ***bloc interfaţă elemente de manipulare (MOVER);*** coordonează logistica la nivelul atelierului;

- ***bloc interfaţă pentru elemente de procesare (PRODUCER*)**; asigură încărcarea sistemelor de control ale staţiilor de lucru şi comandă lansarea execuţiilor.

1.3.7. ARHITECTURA DE REFERINŢĂ CIM-OSA

CIM-OSA (*Open System Architecture*) are ca scop stabilirea unui sistem conceptual bine structurat şi clar constând într-o ahitectură şi o metodologie integrată de proiecare a sistemului de fabricaţie.

Setul de cerinţe care au stat la baza dezvoltării CIM-OSA este:
- pentru o implementare particulară CIM, arhitectura CIM-OSA trebuie să conducă la un "model particular" care să înglobeze informaţiile necesare pentru specificarea unui set de blocuri informatice necesare funcţionării întreprinderii;
- modelul CIM particular trebuie să poată fi executat cu ajutorul tehnologiilor informatice (cerinţa de utilizare a unui set de standarde);
- beneficiarul sistemului CIM trebuie să poată utiliza propria terminologie pentru a putea formula specificaţiile pentru modelul particular (trebuie asigurată independenţa faţă de aspectele de implementare propriu-zisă);
- metodologia porpusă trebuie să asigure suportul pentru un proces etapizat de integrare; nivelul de integrare trebuie să depindă de decizia managerială;
- trebuie asigurată o interconectare informatică dinamică ca alternativă la secvenţe preprogramate de activităţi;
- arhitectura informatică trebuie să poată fi implementată utilizând componente de la diferiţi furnizori;
- sa permită o reconfigurare rapidă hardware şi software.

Structura arhitecturii CIM-OSA

Complexitatea arhitecturii CIM-OSA se justifică prin scopul ei, acela de a defini o structură generică pentru descrierea complet structurată a întreprinderii ca **sistem**. Descrierea include comportamentul dinamic al proceselor de fabricaţie, informaţionale şi de management. CIM-OSA pune un accent deosebit pe definirea unor module informatice, care trebuie să stea la baza proiectării, funcţionării şi mentenanţei sistemului. Tratarea simultană a tuturor acestor aspecte poate asigura consistenţa sistemului „arhitectura particulară CIM " generat.

Principiile arhitecturii de referinţă CIM-OSA sunt :
- abstractizarea;

- modularizarea;
- arhitectura de sistem deschis;
- separarea cerinţelor utilizator de implementarea propriu-zisă;
- utilizarea limbajelor descriptive;
- separarea conducerii sistemului de fabricaţie în raport cu operarea acestuia.

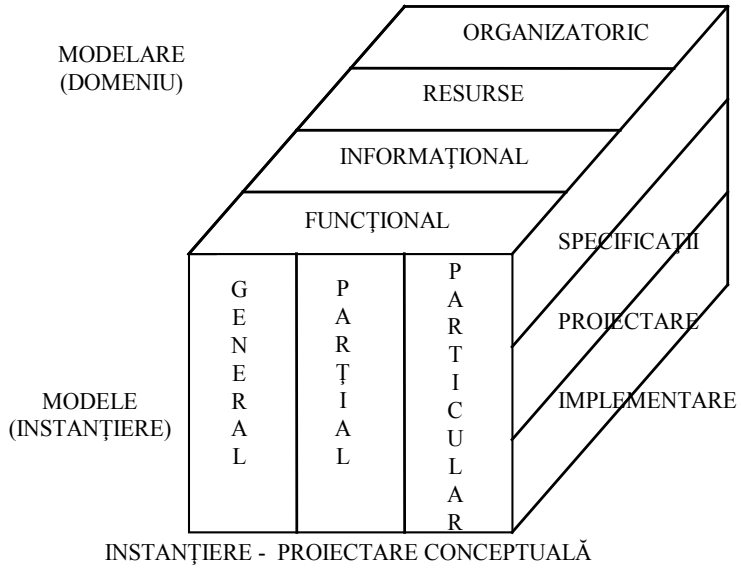


Fig.1.13. Cubul CIM-OSA

Cadrul de modelare CIM-OSA are ***trei nivele de generalitate***, figura 1.13.:

- ***general*** – catalog de referinţe a "blocurilor de construcţie de bază";
- ***parţial*** – seturi de modele parţial instanţiate;
- ***particular*** – soluţia specifică.

Nivelurile ***general*** şi ***parţial*** asigură formularea specificaţiilor de sistem de către beneficiar şi transpunerea lor într-o descriere, respectiv implementare consistentă de sistem CIM.

Nivelul ***particular*** conţine cerinţe pentru operaţii specifice ale întreprinderii reale şi cadrul pentru selectarea şi implementarea componentelor de sistem care satisfac aceste cerinţe (de exemplu: generic–incadrare într-un domeniu, construcţii de maşini; parţial – întreprindere de motoare cu ardere internă; particular – motoare de autoturism).

CIM-OSA oferă o metodologie de ***modelare structurată pe trei nivele***:

- ***definire cerinţe*** - utilizatorul işi defineşte propriile necesităţi;

- ***specificaţii de proiectare***–se structurează şi optimizează cerinţele utilizator şi se asigură interfaţa dintre cerinţa utilizator şi reprezentarea de sistem;
- ***descrierea implementării***–sunt specificate seturile de componente necesare pentru funcţionarea efectivă a întreprinderii.

Ţinând cont de complexitatea sistemului CIM dimensiunea următoare propune patru „***vederi*** *”*, care să permită modelarea independentă a aspectelor majore ale întreprinderii:

- ***funcţional*** – operaţiile efectuate de întreprindere sunt reprezentate ca un set de procese structurate ierarhic; fiecare proces este definit prin evenimentele care-l declanşează, rezultatele obţinute, şi prin descrierea explicită a fluxului de control;
- ***informaţional*** – fluxurile de informaţii şi organizarea ierarhică a tuturor informaţiilor din întreprindere;
- ***resurse*** – totalitatea resurselor de care dispune întreprinderea şi relaţiile cu celelalte structuri (funcţionale, organizatorice, de conducere);
- ***organizatoric*** – se referă la toate informaţiile relevante legate de responsabilităţile în întreprindere.

Comunicaţiile în cadrul sistemului sunt asigurate prin intermediul infrastructurii informatice integratoare, IIS (*Information Infrastructure System*) care oferă serviciile necesare asigurării schimbului de informaţii şi operării.

Conceptul final propus de CIM-OSA pentru integrare îl reprezintă noţiunea de ***entitate funcţională***.

Definiţie: O entitate funcţională este un obiect ideal, independent de implementare, care poate să transmită, să primească, să proceseze şi, opţional, să stocheze informaţie.

O entitate funcţională poate fi un algoritm, o expresie matematică, o funcţie de conducere etc., care descrie o componentă conceptuală a funcţiei de producţie. Entităţilor funcţionale le este asociată noţiunea de ***tranzacţie***, secvenţă completă de apelare, execuţie şi schimb de rezultate între acestea.

Punctele esenţiale ale arhitecturii de referinţă CIM-OSA sunt :

- modelul general se poate reduce la un model particular pe baza utilizării unor biblioteci de module hardware şi software;
- metodele de descriere, configurare şi simulare a sistemului de fabricaţie sunt bine definite;
- rafinarea generic-particular se face prin instanţieri succesive, ceea ce face ca structura să se prezinte consistent şi unitar în fiecare etapă;
- metodologie completă, care se adresează atât beneficiarului, cât şi proiectantului de sistem CIM.

Dificultatea majoră în utilizarea modelului CIM-OSA o constituie faptul că procesul generării arhitecturii particulare este extrem de laborios şi scump.

Modelul general CIM-CASE

Un element comun tuturor proiectelor de implementare a unui sistem CIM este efortul iniţial prin care se încearcă stabilirea unui glosar şi unei taxonometrii comune, ca o precondiţie pentru comunicarea efectivă intre membrii echipei multidisciplinare implicate în procesul de proiectare a fabricii, care este important şi consumă resurse. Din nefericire o parte importantă a rezultatelor acestor eforturi rămân *încapsulate* în baza de cunoştinţe a consorţiului care le-a generat.

Literatura de specialitate propune, într-o fază târzie a CIM [Camarinha,'95], o taxonomie utilizată ca parte a bazei de cunoştinţe pentru un prototip de sistem ierarhic de planificare. În cadrul acestei taxonomii, numai pentru activităţile inginereşti specifice fabricaţiei discrete s-au stabilit peste 157 subactivităţi, organizate pe şase domenii (planificare sisteme, proiectare produs, planificare procese, planificare producţie, planificare execuţie, supervizare). Modelul general al CIM-CASE asigură managementul unui catalog electronic de module software pentru CIM, având ca scop sprijinirea unei generări interactive a unor configuraţii particulare de module dedicate pentru o structură specifică, de exemplu o arhitectură particulară în sensul CIM-OSA.

O *arhitectură particulară* este o configuraţie software pentru un sistem CIM concret**, o *instanţiere*** a modelului de arhitectură de referinţă, cuprinzând un set de

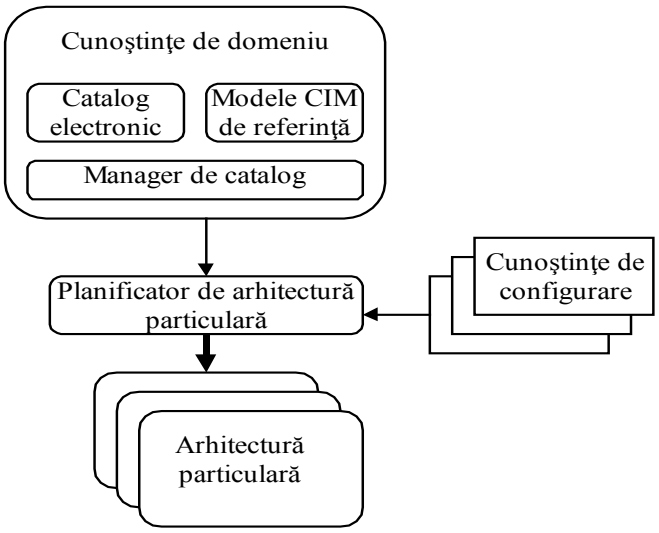


Fig. 1.14. Modul CIM-CASE

module software din catalogul disponibil, integrat prin intermediul unui sistem informatic de suport. Elementul central al modelului este *baza de cunoştinţe* despre

activităţile CIM (taxonometrie şi alte relaţii), aplicaţii de domeniu, taxonomii ale modelelor de date, istoria configuraţiilor şi cunoştinţele de configurare – modele CIM de referinţă. *Managerul de catalog* permite căutarea, explicarea facilităţilor asociate catalogului şi modelelor de referinţă şi asigură platforma de schimb de informaţii care să susţină activitatea echipei interdisciplinare implicată în proiect.

Cea de-a doua componentă funcţională a modelului este *planificatorul de arhitectură particulară,* suport interactiv pentru specificarea şi modificarea arhitecturilor particulare. Modelul este complex şi necesită un efort consistent pentru rafinarea taxonometriei şi pentru precizarea unor relaţii şi dependenţe între activităţi, în afara celor oferite de taxonometrie. Avantajul acestui model poate fi considerat gradul său mare de generalitate şi deschiderea lui către înglobarea unor noi modele şi a experienţei arhitecturilor particulare deja existente.

1.3.8. AVANTAJE ŞI LIMITĂRI ALE PARADIGMEI CIM

CIM reprezintă un număr însemnat de realizări în dezvoltarea sistemelor de fabricaţie, un maxim al orientării "exclusiv" tehnologice dar şi dificultăţi şi limitări.

Analiza universului CIM se va face cu referire la cele cinci paradigme primare [Kidd,94] ce caracterizează fabricaţia:

- paradigma managementului financiar;
- paradigma organizării;
- paradigma controlului;
- paradigma tehnologică;
- paradigma proiectării.

Din punct de vedere al managementului financiar contribuţia CIM se poate încadra în obţinerea de *beneficii tangibile şi intangibile* [Kidd,94].

Dintre ***beneficiile tangibile***, al căror impact este măsurabil direct prin parametrii de performanţă financiară ai sistemului, se pot preciza:

- ***diminuarea stocurilor interoperaţionale şi de produse finite***; flexibilitatea proceselor, optimizarea fluxurilor de produse, creşterea calităţii (reflectată în mai puţine retururi) şi perfecţionarea ordonanţării determină o micşorare a „producţiei în curs"(*work in progress*) şi la diminuarea stocurilor;
- ***utilizarea optimă a spaţiului existent***; concentrarea pe tehnologie, înlocuirea maşinilor clasice cu maşini cu comandă numerică, centre de prelucrare, sisteme flexibile de fabricaţie ceea ce aduce beneficii legate de utilizarea intensivă a spaţiilor de producţie disponibile, micşorarea costurilor de transport şi depozitare interoperaţională, al costurilor totale de întreţinere;

- ***creşterea calităţii***; tehnicile de control de calitate şi atenţia acordată acestei dimensiuni conduc la micşorarea retururilor şi la diminuarea costurilor de întreţinere şi service în garanţie precum şi la diminuarea producţiei care nu satisface cerinţele impuse.

O problemă importantă este cea a ***beneficiilor intangibile*** şi a faptului că efectul lor este indirect, determină creşterea profitului dar nu au un impact imediat, cuantificabil direct, asupra costurilor de producţie. Dintre acestea se pot preciza:

- creşterea flexibilităţii;
- creşterea vitezei de răspuns la cerinţele pieţii;
- micşorarea timpului de fabricaţie a unui produs.

Încă din fazele de început ale CIM şi SFF s-a pus problema stabilirii unor tehnici care să permită cuantificarea beneficiilor intangibile [Kidd,94].

Modelul de preţ de cost propus ia în calcul atât modelul economiei de scară cât şi cel al ***"economiei de scop"***, (figura 1.12.) cu precizarea că principiile *economiei de scop* şi cele ale *economiei de scară* nu se pot aplica strict, întreprinderile folosind cele doua modele, sau combinaţii între ele în funcţie de condiţiile specifice pieţii.

Definiţie. ***Economia de scop*** este abilitatea de a aloca şi realoca dinamic capitalul fix, în funcţie de scopul curent al întreprinderii [Kidd,94].

Din punct de vedere al structurii organizatorice şi al controlului, experienţa CIM se reflectă în:

- structuri ierarhice care determină, din considerente de integrare, un flux important de date comandă şi control;
- dezvoltarea de ***arhitecturi de control de referinţă*** şi de ***arhitecturi de referinţă*** ca metodologii de implementare a fabricii integrate prin calculator.

Ca deficienţe se pot sublinia:

- conceptul de CIM poate ***fi dezvoltat numai la nivelul tuturor cerinţelor funcţionale simultan***; structura ierarhică de sistem necesită generarea unui număr foarte mare de interfeţe specializate între diversele componente; costul modificărilor este prohibitiv;
- implementarea CIM „***rigidizează***" procedurile şi structurile existente într-o companie;
- sistemele CIM impun conducerea în timp real, planificarea, controlul şi supervizarea ***centralizat ierarhică*** a tuturor proceselor operaţionale.

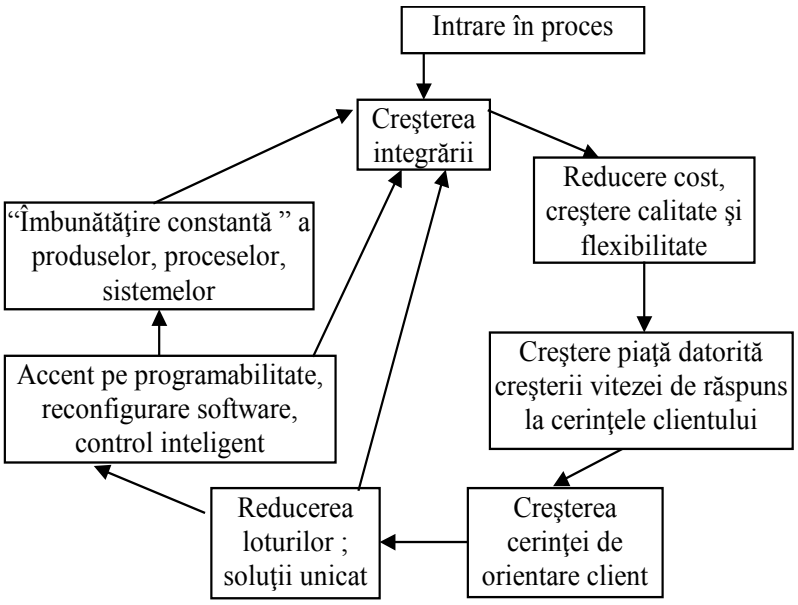


Fig. 1.15. Modelul economiei de scop

Din punct de vedere tehnologic

CIM pune în discuţie problema foarte importantă a integrării întregii activităţi a companiei, de la marketing şi vânzări, la proiectare, planificare producţie, fabricaţie, cercetare-dezvoltare, management financiar, asigurarea calităţii. Prin prisma acestei paradigme se pot enumera ca beneficii:

- dezvoltarea sistemelor şi protocoalelor de comunicaţii;
- dezvoltarea tehnologiei informatice şi a bazelor de date;
- controlul computerizat;
- automatizarea procesului de proiectare, planificare producţie, fabricaţie, depozitare şi dezvoltarea de platforme funcţionale de integrare activităţi, cu implicaţii importante în extinderea spre întreprinderea virtuală;
- sistemele CIM integrează managementul de procese şi aplicaţiile CAD prin module software care permit accesul la aceeaşi bază comună de date; modelele de date partajate şi accesul distribuit la acestea sunt elementele cheie care au permis crearea platformelor de sistem distribuit şi ierarhizat;
- implementarea CIM în fabrica viitorului, prin promovarea tehnologiilor informatice şi de comunicaţii, va asigura un sistem informatic şi de navigaţie flexibil şi eficient pentru agenţii autonomi .

Ca deficienţe se pot sublinia:

- ***focalizarea excesivă pe cerinţe funcţionale***: proiectare, planificare, fabricaţie neasigurând o integrare reala a „procesului de afaceri" incluzând marketingul, cercetarea/dezvoltarea etc; **M** de la CIM înseamnă cu ***adevărat „numai fabricaţie"*** (*Manufacturing*) [Warneke,93];
- ***încrederea excesivă în posibilităţile oferite de computer;***
- focalizarea asupra proceselor de optimizare matematică;
- incertitudinea în diagnosticare;
- automatizarea proceselor ***nu „exploatează" creativitatea umană*** – orientarea excesiv spre tehnologie;
- investiţia în tehnologie se referă accentuat la reducerea costurilor şi creşterea productivităţii fără a ţine cont de implicaţiile de ansamblu, socio-culturale şi motivaţionale asupra resursei umane.

Din punct de vedere al paradigmei proiectarii trebuie subliniate deficienţele majore ale proiectării tradiţionale preluate şi promovate de CIM :

- ***abordare reducţionistă***; optimizarea la nivel de întreprindere, pe baza optimizării la nivelul de activitate elementară;
- ***mono-disciplinaritate*** sau în cel mai bun caz multi-disciplinaritate dar nu inter-disciplinaritate;
- ***inginerie serială***; în fazele târzii ale CIM ingineria concurentă reprezintă o presiune foarte puternică, care va conduce la o schimbare esenţială a paradigmei proiectării.

CIM, aşa cum a fost dezvoltat drept concept, reprezintă un sistem închis. Concentrarea s-a făcut asupra rezolvării aspectelor tehnice conducând la izolarea de aspecte legate de integrarea resursei umane, aspecte sociale şi organizatorice. Tendinţa de reducere a rolului resursei umane în proiectarea tehnologică a determinat automatisme şi de-specializări, cu implicaţii majore asupra satisfacţiei în muncă şi a creativităţii umane. Voluntar sau nu, CIM a promovat, prin accentul asupra tehnologiei, transferul deciziei către calculator şi ignorarea potenţialului uman.

Rolul CIM nu poate fi minimalizat, iar noile paradigme legate de viitorul sistemelor de fabricaţie reiau un mare număr de probleme puse de CIM şi propun soluţii compatibile conceptului integrat fabricaţie orientată ***tehnologic-antropocentric-organizatoric***.

CONCLUZII

Capitolul 1 îşi propune abordarea întreprinderii industriale ca sistem dinamic, deschis, în interacţiune cu mediul în care funcţionează. Prezentarea este motivată de necesitatea analizei contextului tehnologic, social şi organizatoric propus de paradigmele de fabricaţie tradiţională şi mai apoi CIM, ca etapă esenţială în stabilirea modelelor fabricii viitoare.

Încă de la începutul capitolului,i se motivează necesitatea transformării unei întreprinderi într-un ***sistem cu învăţare*** *şi se propune un model generic al sistemului de fabricaţie integrat în mediu, care poate susţine repetarea continuă a ciclului* ***sesizează-decide-acţionează*** *sub diferite orizonturi de timp.*

Orientarea tehnologică, manifestată începând din 1973, an în care se promovează pentru prima dată termenul de ***Computer Integrated Manufacturing*** *(dar nu încă acronimul CIM) este prezentată orientat pe platforme funcţionale pentru integrare activităţi şi arhitecturi de referinţă. Se propune o încadrare a diverselor module funcţionale într-o „****galaxie CIM****”.*

Pentru a modela întregul ciclu de viaţă al sistemelor de fabricaţie se studiază arhitecturile dezvoltate sub paradigma CIM. Se face o analiză a rolului arhitecturilor de referinţă şi se propune încadrarea lor în două clase, care se referă ***la proiectarea dispunerii structurale a sistemului fizic*** *şi* ***organizarea dispunerii structurale a dezvoltării şi implementării unui proiect sau program de tipul întreprindere integrată****.*

Pentru fiecare dintre aceste clase se prezintă modelele de referinţă. Pentru prima clasă de arhitecturi este prezentată arhitectura ierarhică de control organizată pe patru niveluri. Se propune un model de sistem de control al activităţilor de producţie posibil de instanţiat în oricare din nodurile ierarhiei şi se propune o soluţie bloc pentru nivelul celulei de fabricaţie. Se face o asociere între fazele ciclului de viaţă al întreprinderii CIM şi niveluri de organizare ierarhică a întreprinderii. Pentru cea de a doua clasă de arhitecturi se prezintă arhitectura de referinţă CIM-OSA, care se constituie într-o metodologie integrată pentru proiectarea şi operarea fabricii CIM.

În încheiere se propune o analiză critică a soluţiei CIM orientată pe cele cinci paradigme care stau la baza unui sistem de fabricaţie: paradigma managementului financiar, paradigma organizatorică, paradigma controlului, paradigma tehnologică şi paradigma proiectării. Sunt prezentate principalele beneficii ale CIM precum şi ***limitările orientării tehnologice****.*

Capitolul 2 – cuprins

2.1. Cerinţe pentru sistemul viitor de fabricaţie **45**

2.2. Vectori de schimbare **48**

2.2.1. Modelul sistemului holonic de fabricaţie **48**
- **Prototip de sistem holonic de fabricaţie** **51**
- **Convergenţa conceptuală: sistem holonic-întreprindere virtuală** **55**

2.2.2. Model de sistem fractal de fabricaţie **56**
- **Navigaţia şi controlul fractalilor** **59**
- **Model de planificare producţie şi control de tip fractal** **60**

2.2.3. Model de sistem bionic de fabricaţie **62**

2.2.4. Comparaţie între modelele holonic / fractal / bionic **64**

2.3. Flexibilitatea sistemelor de fabricaţie **67**

Concluzii **70**

2

SISTEMUL DE FABRICAŢIE POST-CIM

2.1. CERINŢE PENTRU SISTEMUL VIITOR DE FABRICAŢIE

Indiferent că termenul utilizat este ***NGMS*** (*Next Generation Manufacturing Systems*), ***ADMS*** (*Advance Manufacturing Systems*), ***DEE*** (*Dinamic Extended Enterprise*), ***IMS*** (*Intelligent Manufacturing Systems*) sau ***CE*** (*Concurrent Enterprise*) având sprijinul oferit de noile modele de referinţă ale fabricii (holonic, fractal, bionic, virtual), sistemul viitor de fabricaţie va fi rezultatul presiunii exercitate de dezvoltarea tehnologiilor informatice şi de comunicaţii în medii electronice virtuale şi vor avea ca elemente comune caracteristice ***autonomia***, ***co-operarea***, ***interdisciplinaritatea***, ***abordarea integrată produs-proces-sistem***, ***reconsiderarea resursei umane***, va fi ***agilă în organizare*** şi ***virtuală ca asociere şi distribuţie geografică***.

Schimbările tehnologice dar şi cele de mentalitate înregistrate în utilizarea informaţiei în spaţiul de reinginerie a afacerii (*bussiness re-engineering*), care are ca nucleu comerţul electronic, repun în discuţie esenţa schimbării organizatorice a întreprinderii compatibile cu noile meta-modele [Palmer,96]. Comunicaţiile intercompanii apelează o clasă nouă de servicii, superioare calitativ formei convenţionale EDI (*Electronic Data Interchange*), denumită Internet-EDI. Tranzacţiile între parteneri se realizeaza „aproape în timp real”.

Cercetările orientate Internet-EDI subliniază următoarele caracteristici:

- tranzacţiile Internet-EDI sunt rapide, oferă securitate ridicată (formatul este Internet-like) având costuri mai reduse decit EDI, în primul rând prin utilizarea unei infrastructuri publice de comunicaţii;
- internetul poate fi adecvat comerţului electronic datorită mediului de comunicaţii aliniat la standardul de sistem deschis de comunicaţii OSI şi datorită disponibilităţii (extinderii reţelei şi creşterii numărului de ofertanţi de servicii, *service-providers*);
- ***„piaţa virtuală”,*** spaţiul comerţului electronic, este o idee în curs a deveni realitate, dar care necesită, alături de perfecţionarea proceselor orientate

tranzacţie, o nouă legislaţie, intensificarea comunicaţiilor inter-organizaţii, organizarea întreprinderilor orientat pe colaborarea intensivă.

Spaţiul Internet-EDI este legat de paradigma managementului financiar, dar şi de paradigma tehnologică. Mediul electronic virtual sprijină cerinţa de afaceri, de a fi „mai rapid pe piaţă" (*time to market*) şi susţine operarea companiei în limitele unui „inventar zero"(*Just in-Time Inventory System*).

Din punctul de vedere al paradigmei organizatorice noul cadru exterior companiei impune înlocuirea „integrării verticale a întreprinderii", bazată pe tehnologii informatice şi structuri ierarhice de control aferente CIM prin forme organizatorice elastice, de tip colaborativ, caracterizate prin autonomie şi coordonare multi-agent. Dacă sistemele CIM s-au dezvoltat ca urmare a unui mediu plin de incertitudini şi riscuri create de comportamentul partenerilor de afaceri, într-o piaţă care devenise un coordonator prea lent şi costisitor al producţiei diferitelor companii cuplate prin sistemul tradiţional de afacere, noile facilitaţi de integrare în reţele externe rapide de comunicaţii şi comerţ electronic, pot reduce gradul de incertitudine şi risc între parteneri prin operarea în timp real.

Din punctul de vedere al paradigmei organizatorice este esenţial ca noile sisteme de producţie trebuie să fie mult mai elastice, locul „***piramidei organizatorice"*** fiind luat de „***casa organizatorică"*** [Kidd,94].

Paradigma tehnologică propune dezvoltarea tehnicilor de inteligenţă artificială, a inovaţiei în reţea, a bazelor distribuite de cunoştinţe capabile să suţină eforturile echipelor virtuale.

Din punct de vedere al paradigmei proiectării se evidenţiază presiunea Ingineria Concurentă (CE–*Concurrent Engineering*) şi a noii tendinţe, corelată cu globalizarea pieţii şi concentrarea companiilor pe „competenţe specifice" (*core competence*), de creare de reţele de capitalizare a cunoştinţelor.

Toate conceptele propuse au ca obiectiv adaptarea meta-sistemului întreprindere la schimbările dinamice, puţin predictibile, ale mediului specific de afaceri intern şi extern sistemului de fabricaţie.

Noua întreprindere va avea funcţiuni care imită organismele vii (auto-organizare, auto-refacere, creştere şi evoluţie), va utiliza modele şi va propune metodologii care să acceseze toate nivelurile procesului de afaceri, va răspunde rapid-orientat client la cerinţele pieţii, se va armoniza cu mediul (ecologic) şi va deţine informaţii sistematizate relativ la întregul ciclu de viaţă al produsului obţinut în întreprinderea virtuală (VE-Virtual Enterprise)

Răspunsurile generice la problema creării unor structuri organizatorice pentru întreprinderea integrată se bazează pe principiul următor***: funcţionalitatea sistemului de producţie al fabricii*** (fabrica privită ca nucleu de piaţă globală şi

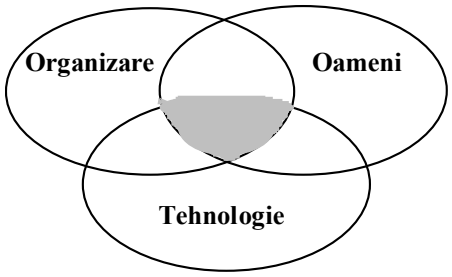
flexibilă ca mod de autoorganizare) ***trebuie să se subordoneze scopului, obiectivelor şi metricii de evaluare a performanţelor de sistem de afaceri ca supra strat (shell) al strategiei CIM.***

O paradigmă sustenabilă pentru fabricaţie are la bază conceptul că soluţia tehnologică (de exemplu: produsele şi procesele asociate de-a lungul ciclului de viaţă) trebuie să răspundă cerinţelor pieţei şi să fie în conformitate cu constrângerile şi resursele disponibile [BIBA, 95].

Crearea unui lanţ de valoare adăugată de fiecare partener din afacere, tratată ca mediu dinamic holistic, conduce la ideea creării unui parteneriat, în cadrul VE, şi distribuirii riscului între parteneri în locul unor relaţii strict focalizate comandă-contract clasice.

Fabrica viitorului propune concepte ca:

- ***fabrica „integrată prin oameni şi calculator (HCIM)”*** [Warneke, 93];
- ***fabricaţie balansată*** (între tehnologie, organizare şi oameni- „templul” propus de fabricaţia agilă) şi abordare interdisciplinară a universului fabricaţiei;



- ***integrarea oamenilor*** (oameni care comunică şi cooperează);
- ***integrarea om-computer*** (oameni interacţionând prin intermediul calculatorului);
- ***integrarea tehnologică*** (maşini interfaţate cu maşini);
- ***integrarea şi implicarea oamenilor*** în ***fabricaţie*** cu scopul de a impune modul participativ şi nu cel intervenţionist al oamenilor în sistemul de fabricaţie.

Paradigma proiectării propune ***metodologia interdisciplinară de proiectare*** prin utilizarea ingineriei concurente în locul celei seriale.

Proiectarea întreprinderii viitoare pote fi caracterizată de:

- abordare holistică a sistemului;
- aplicarea conceptelor CE la proiectarea sistemului de fabricaţie;

– orientarea organizării şi oamenilor, împreună cu cea tehnologică, la fiecare etapă din ciclul de viaţă al întreprinderii.

Paradigma proiectării noii întreprinderi se bazează pe modele de proiectare iterativă, care să asigure realizarea sistemului dorit prin simulare în mediu virtual, prototipizarea rapidă şi simulare, iar ca model de proiectare se poate considera modelul spiral al ciclului de viaţă, dezvoltat iniţial pentru ingineria software [Boehm,89] şi adaptată la proiectarea întreprinderii [Kidd,94].

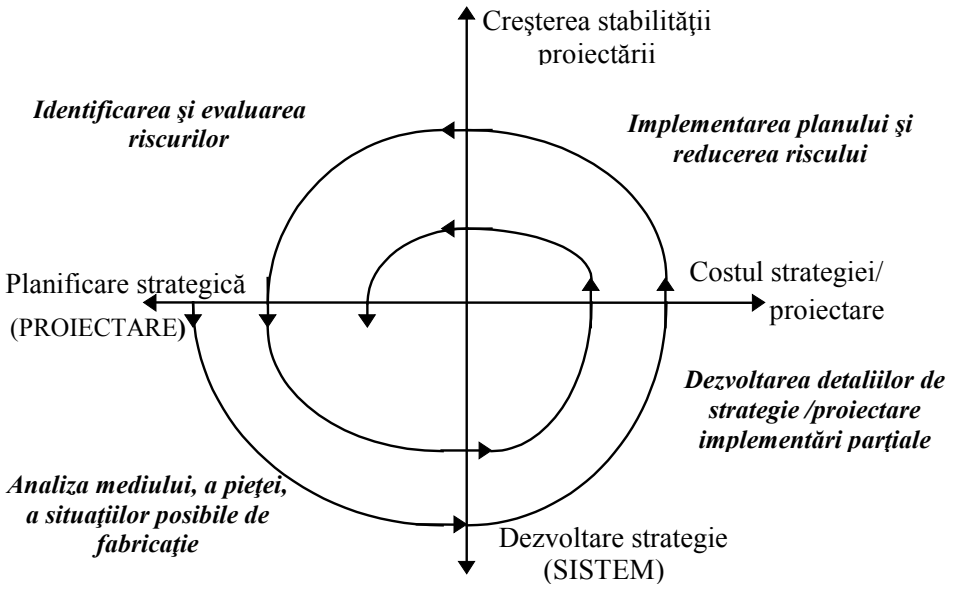


Fig. 2.1. Modelul spiral al ciclului de viaţă al întreprinderii

2.2. VECTORI DE SCHIMBARE

2.2.1 MODELUL SISTEMULUI HOLONIC DE FABRICAŢIE

Sistemele holonice îşi au originea în teoria filozofică asupra creării şi evoluţiei sistemelor complexe adaptive – sisteme sociale, biologice etc.). În literatura de specialitate se afirmă că acest tip de sistem are o bază mai solidă decât alte abordări deoarece filozofii, cei care au dezvoltat teoria holonilor, **explică,** nu doar **observă** fenomenele.

Noţiunea de holon a fost introdusă de Arthur Koestler. Termenul de holon este o combinaţie între cuvintul grec *holos* (întreg) şi sufixul *on*, care sugerează o particulă, o parte (prot-on, neutr-on), [Brussel,95].

Sursa de inspiraţie pentru Koestler au constituit-o două observaţii:

- ***ierarhiile dinamice în sistemele bazate pe celule (sisteme vii) şi sistemele sociale;*** Sistemele complexe se formează şi evoluează mai rapid dacă sunt compuse din forme/subsisteme intermediare stabile; în caz contrar, sistemul complex rezultat va fi un sistem rigid ierarhic. Această caracteristică a sistemelor complexe este ilustrată, peste tot în literatura de specialitate, de parabola celor doi constructori de ceasuri, lansată de laureatul premiului Nobel – Herbert Simon – ceasornicarul care realizează ceasuri utilizând macro-subansamble stabile cîştigă competiţia cu cel care construieşte ceasuri prin montarea element cu element, caz în care, orice situaţie neprevăzută poate conduce la reluarea întregului ciclu de asamblare.
- ***nivelul de generalizare parte, faţă de întreg, este subiectiv şi nu este unic.*** Studiul formelor intermediare stabile din organismul viu sau din domeniul social, structurate ierarhic, sprijină concluzia că **întregul şi partea** nu există în mod absolut, descompunerile nu sunt unice şi depind de abordare, sunt subiective.

Prin introducerea conceptului de Sistem Holonic de Fabricaţie (**SHF**) s-a urmărit extinderea, prin similitudine, pentru fabricaţie, a avantajelor conceptual sistemice pe care noţiunea de holon le asigură pentru organismele vii şi sistemele sociale:

- **stabilitate** referitor la perturbaţii;
- **adaptabilitate** relativ la schimbări externe;
- **flexibilitate -** capacitatea de autoconfigurare internă;
- **eficienţă** în utilizarea resurselor.

Conceptul de **SHF** combină trăsăturile sistemelor ierarhice şi respectiv heterarhice, ceea ce permite conservarea stabilităţii oferite de structurile ierarhice în condiţiile asigurării simultane a flexibilităţii dinamice a structurilor heterarhice.

Descompunerea heterarhică, îşi propun eliminarea deficienţelor descompunerii ierarhice. Conceptul propune modularizarea prin încapsularea informaţiei în „obiecte" relativ autonome. În locul unor relaţii de tip master-slave, specifice sistemelor ierarhice, descompunerea heterarhică propune o structură de comunicaţii care să permită modulelor unui sistem să comunice direct între ele, ca noduri. Sistemele organizate heterarhic funcţionează ca procese (agenţi) autonome, care cooperează, fără a avea un control centralizat. Deciziile sunt luate în urma negocierilor, prin acord mutual, iar informaţiile sunt schimbate liber între agenţii autonomi.

Avantaje ale arhitecturilor heterarhice de control *:*

- *modularitatea naturală* a sistemului stă la baza definirii agenţilor autonomi*;*
- *flexibilitate dinamică;*
- *adoptarea deciziei* pe bază de negociere*;*
- *disfuncţiile* manifestate la nivelul unui agent autonom influenţează limitat performanţele sistemului.

Dezavantaje ale arhitecturilor heterarhice de control:

- *volumul mare de informaţie* schimbată între ageţii autonomi;
- *posibilitatea adoptării de decizii inconsistente* datorită pierderii informaţiilor globale.

Definitii:

Holonul: *Un bloc autonom şi co-operativ al sistemelor de fabricaţie, incluzând uzual o componentă de procesare a informaţiei şi o componentă de procesare fizică. Poate asigura transformarea, transportul, memorarea şi/sau validarea informaţiei sau a obiectelor fizice şi poate face parte din alt holon.*

Autonomie*: Capacitatea unei entităţi de a crea, controla şi executa propriul plan conform unei strategii elaborate pentru orizontul de timp mediu şi mare.*

Co-operarea*: Este procesul în care un set de entităţi dezvoltă şi execută planuri mutual acceptate, pentru îndeplinirea unui scop prestabilit.*

Holarhia

- *Un sistem de holoni care co-operează pentru atingerea unui scop sau a unor obiective comune, pe baza unor reguli care guvernează cooperarea între aceştia şi respectiv limitează autonomia lor [Brussel,1995].*
- *Un sistem ierarhic de holoni în care fiecare dintre aceştia funcţionează:*
  - *ca un întreg autonom, relativ la componentele sale;*
  - *ca o unitate dependentă, în relaţie de subordonare faţă de nivelurile ierarhice de control;*
  - *în interdependenţă cu mediul său local.*

SHF: *Sistemul Holonic de Fabricaţie este o holarhie, care integrează totalitatea activităţilor de fabricaţie, de la gestiunea comenzilor până la proiectare, producţie şi marketing, pentru a realiza o întreprindere de producţie agilă.*
Atribute holonice: *Setul minim de atribute care, asociate unei entităţi, permit transformarea ei într-un holon. Setul minim îl constituie autonomia şi co-operarea.*

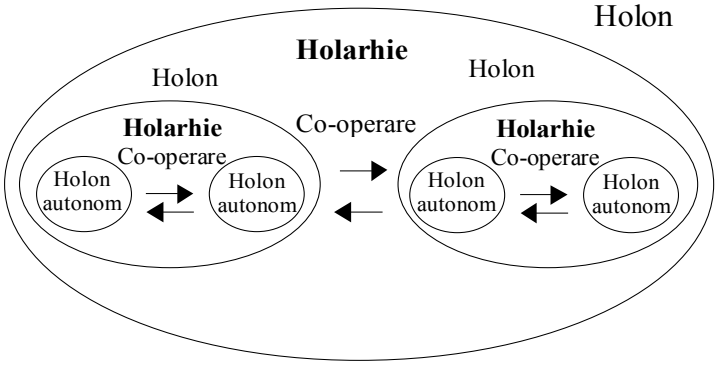


Fig. 2.2. Holarhie şi conceptul de co-operare

Comportarea holonică poate fi exprimată prin două tendinţe complementare, „**tendinţa de auto-afirmare**" şi „**tendinţa de integrare**" [Brussel,1993], [Brussel,94]. Stabilitatea holonilor rezultă din abilitatea lor de a acţiona autonom, fară o asistenţă continuă, de la nivel superior, în situaţii neprevăzute.

Autonomia holonilor de a executa task-uri date, este o proprietate moştenită şi exprimă „tendinţa de auto-afirmare" a acestora. Ea se referă la: ***controlul local şi operarea maşinilor, optimizarea la nivel local, auto-ordonantare, auto-configurarea, auto-diagnoza, auto-învăţarea, auto-repararea*** etc.

Co-operarea prin negociere între holoni exprimă „tendinţa de integrare", ce este datorată nevoii de supravieţuire şi evoluţie în mediu şi se referă la proprietăţi ca: ***flexibilitate, orientare-scop, auto-organizare, extensibilitate***.

Fiecare unitate de producţie (resursă tehnologică, capacitate de producţie) poate fi un holon. Aceste unităţi cooperează între ele, de la planificare şi ordonare până la prelucrări fizice directe, pentru fabricarea unui produs. Structura sistemelor holonice de fabricaţie are la bază o ierarhie funcţională şi unităţi auto-similare (o structură repetabilă la toate nivelurile), figura 2.3., iar ca suport informatic proiectarea orientată pe obiecte.

Prototip de sistem holonic de fabricaţie

Pentru evaluarea capacităţii de reprezentare a conceptului de holon a fost dezvoltat un SHF la nivelul unui atelier [Brussel,95]. Prototipul propus de holarhie controlează un atelier format din staţii robotizate de asamblare, interconectate printr-un sistem automat de transport. Structura sistemului, denumit sistem Spartan, este minimă, fiind prima etapă în dezvoltarea unui sistem agil de fabricaţie. Prototipul de holarhie este totuşi generic relativ la procesele de fabricaţie şi poate sta la baza configurării altor sisteme de fabricaţie.

Au fost stabilite patru tipuri specifice de holoni: **holonul produs**, **holonul comandă, holonul resursă şi holonul personal**. Holonii definiţi sunt componente IT active, agenţi autonomi, fiecare dintre ei acţionând ca serveri pentru celelalte componente ale holarhiei.

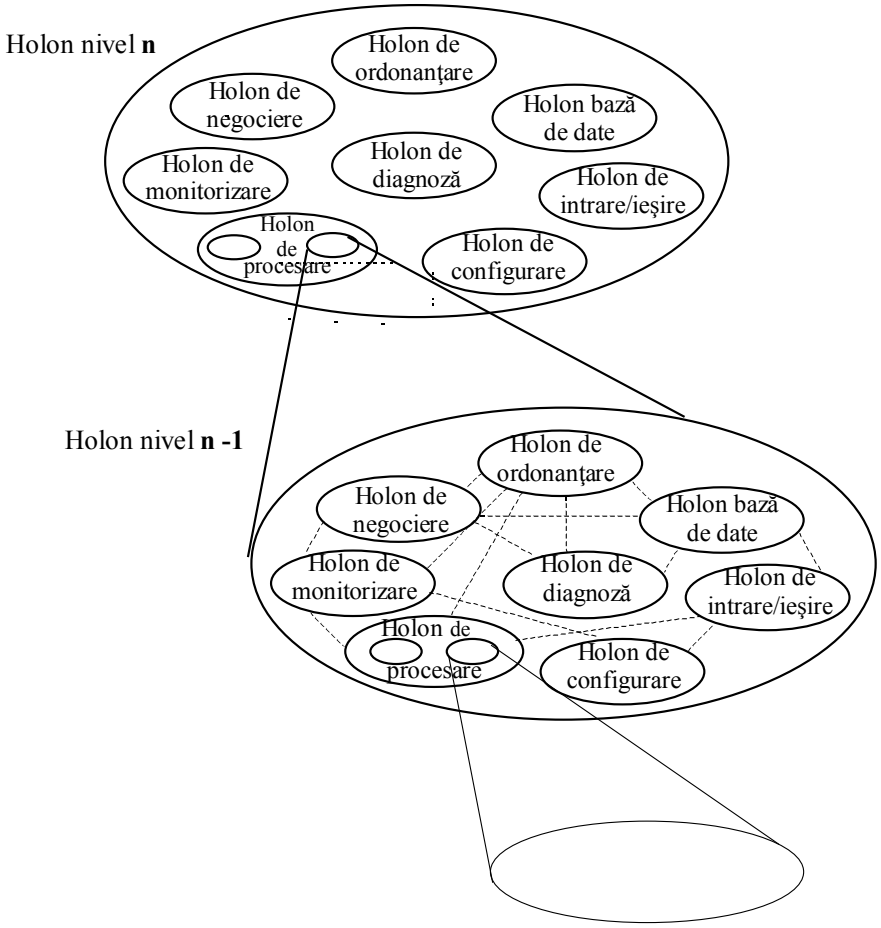


Fig.2.3. Ierarhie funcţională de holoni

Holonul produs este asociat fiecărui produs, conţine toate informaţiile curente referitoare la: ciclul de viaţă al produsului, cerinţe utilizator, proiectare şi planuri de proces şi are implementate mecanisme care să-i permită generarea de planuri alternative de procesare atunci când maşina/dispozitivul devin indisponibile din motive neprevăzute. Holonul produs înglobează *cunoştinţele despre proces şi produs* necesare stabilirii corecte a instanţelor produsului, cu asigurarea calităţii.

Abordarea este similară celei din ingineria concurentă, bazată pe date şi cunoştinţe despre: produs, proces (tehnic) şi sistem (facilităţi de fabricaţie).

Holonul resursă controlează o resursă de producţie din sistemul holonic de fabricaţie. El poate reprezenta o resursă elementară (maşină cu comandă numerică, robot, paletă, dispozitiv, sculă) sau un ansamblu de resurse (statie de lucru, atelier, secţie etc.) şi conţine metodele necesare rezervării şi alocării resurselor de producţie, cunoştinţele şi procedurile pentru organizarea, utilizarea şi controlul acestora în procesul de fabricaţie.

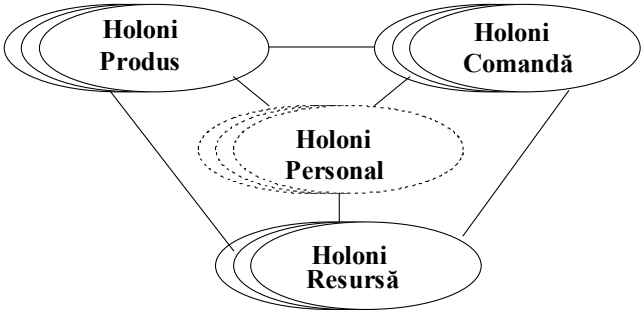


Fig. 2.4. Tipuri de holoni

Holonul comandă încapsulează un ordin de producţie. El reprezintă o entitate având răspunderea realizării activităţilor la timp şi corect. Holonul comandă capturează toate informaţiile referitoare la un job (sarcină de lucru, comandă de fabricaţie).

Holonul personal, a fost introdus ulterior, ca esenţă a dezvoltării antropocentrice viitoare a sistemelor de fabricaţie. Acest holon manifestă orientarea resursă umană prezentă în toate implementările post-CIM şi oferă informaţii şi servicii suplimentare, care sprijină holonii de bază în luarea deciziei.

Holonul produs este generat de nevoile pieţei. Pe baza cererii curente şi a setului dinamic de holoni **resursă**, se stabileşte modul în care va fi realizat produsul. Holonii **comandă** sunt generaţi de comenzile beneficiarilor şi negociază cu holonii **produs** şi holonii **resursă,** relevanţi produsului. Holonul **comandă** acordă atenţie aspectelor logistice şi declanşează schimbul de informaţii tehnice dintre holonul **produs** şi holonul **resursă.**

Arhitectura holonului „**controler de maşină unealtă**" are la bază două concepte legate de autonomie:

- autonomie faţă de nivelurile inferioare (*downstream autonomy*); se utilizează notiunea de **„driver de dispozitiv**";
- autonomie faţă de nivelurile superioare (upstream autonomy); se utilizează noţiunea de **„trasături de fabricaţie"**[Brussel,93].

Conceptul de **„driver de dispozitiv”** a fost definit şi utilizat iniţial, pentru integrarea echipamentelor de comandă a roboţilor industriali, în aplicaţii care presupuneau programarea şi coordonarea acţiunii mai multor roboţi. Diversitatea limbajelor de programare a acestora şi necesitatea programării off-line au fost principalele argumente care au condus la dezvoltarea unei „interfeţe soft neutrale” denumită „driver de dispozitiv ”.

„Driver-ul de dispozitiv” dezvoltat pentru NC are scopul de a asigura atributele holonice necesare transformarii NC clasic , într-un sub-holon NC. Indiferent cât este de „rigidă” soluţia tehnologică iniţială a NC, în care intervenţia şi comunicarea cu lumea exterioară este „aproape imposibilă” sau se poate realiza numai prin soluţii particulare, nestandard, sub-holonul NC astfel creat, asigură independenţa faţă de cod standard ISO pentru programul piesă şi permite coordonarea dintre sub-holonii din holonii resurse. Noţiunea de „driver de dispozitiv” pentru NC are ca punct de pornire noţiunea dezvoltată iniţial pentru roboţi dar se deosebeşte semnificativ faţă de aceasta prin specificaţiile impuse, care nu sunt legate numai de crearea unei interfeţe software care să asigure interpretarea-compilarea unui „limbaj neutral”(de exemplu: NIRO-STEP pentru roboţi) ci are scopul de a implementa atributele holonice , unei structuri NC clasice. Notiunea de “driver de dispozitiv” poate fi asociată soluţiei de ***OpenCNC*** *(„software NC” [Fall,95]), pentru generarea eficientă a holonului NC.*

„Fabricaţia bazată pe trăsături” asigură autonomia sub-holonului NC faţă de nivelurile superioare. Cerinţele schimbului de fişiere produs între holonii din SHF pot fi asigurate prin utilizarea tehnologiei orientată trăsături. Fiecare holon din sistem dispune de flexibilitatea de a utiliza tipul specific de trăsături; holonul proiectare, holonul ordonanţare, holonul de planificare procese şi holonul maşină utilizează diferite niveluri de abstractizare a datelor asociate trăsăturilor aceleiaşi entităţi-produs.

Holonul NC primeşte un program constând în descrierea trasăturilor de fabricaţie a piesei solicitate şi nu un program explicit, care să impună o succesiune precisă de operaţii. Holonul NC urmează să decidă autonom asupra sevenţelor de operaţii, care decurg din ordinul generic de fabricaţie orientat trăsături, rezultând: secvenţa de operaţii tehnologice, dispozitivele utilizate, parametrii de prelucrare, etc.

Auto-diagnoza şi auto-repararea sunt caracteristici importante pentru SHF. În Holonul NC sunt implementate două modele de diagnostic, prin tratarea on-line respectiv off-line a informaţiei senzoriale. Informaţia senzorială este utilizată pentru auto-diagnoză contribuind la creşterea autonomiei controlerului NC.

Algoritmul propus pentru task-ul de control este bazat pe conceptul de operaţii ne-întreruptibile (*Non-Interruptable Operations*, NIO).

Pentru „driverul de dispozitiv” au fost identificate:

- clasa **„capabilităţi de observare”;** este formată din task-uri care urmăresc viteza de rotaţie, rata de alimentare, starea transmisă de task-ul original de control al NC-ului, executarea programului piesă etc;
- clasa „**acţiuni imediate”;** este formată din comenzi tip actiune pentru executarea operaţiilor de prelucrare sau de tip reacţie la perturbaţii şi stă la baza implementării conceptului de NIO.

În literatura de specialitate se propune arhitectura de sistem de control de tip „architectura subînsumată”, adaptată la domeniul sistemelor holonice [Brussel,94] şi implementată pentru holonul NC, figura 2.5.

Arhitectura este compusă dintr-un număr de niveluri, corespunzător specificului proceselor de prelucrare mecanică. Nivelurile sunt aranjate în paralel, fiecare nivel reprezintă o funcţiune independentă şi răspunde de un singur task.

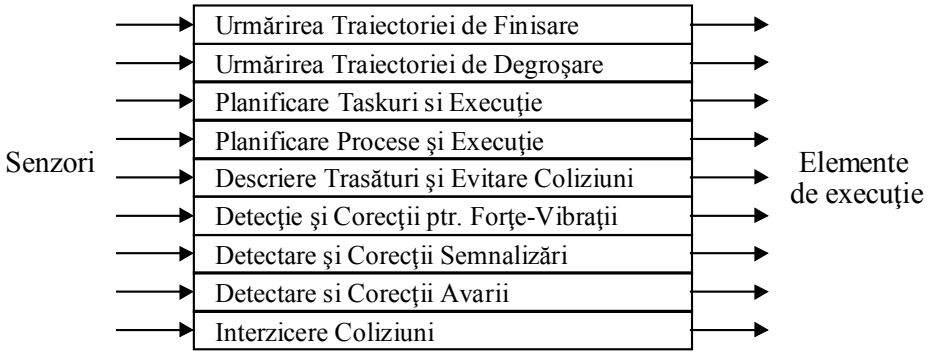


Fig. 2.5. Arhitectura subînsumata pentru sistem holonic

La un moment dat este activă numai una dintre legile de control, în funcţie de regim şi de starea sistemului.

- clasa funcţiilor **„mod de control”**

Convergenţa conceptuală:
Sistem holonic de fabricaţie - Întreprindere Virtuală

Prototipul de holarhie nu este limitat la nivelul întreprinderii şi este compatibil cu conceptul de întreprindere extinsă. Aceasta, îmbinând anvantajele sistemelor ierarhice cu ale celor heterarhice, asigură planificarea şi controlul dar şi susţinerea interacţiunii şi integrării pe orizontală.

Holonul comandă nu este restricţionat la nivelul limită al unei singure fabrici. Datorită proprietăţii de auto-similitudine „macro” holonul comandă beneficiar, împreună cu aspectele financiare asociate se poate descompune în holoni comandă distribuiţi în mai multe fabrici, dacă acest lucru este necesar.

Holonul produs poate depăşi, de asemenea, graniţele unei singure fabrici ceea ce este în corelaţie cu conceptul de integrare a lanţurilor logistice client – furnizor şi cu noţiunea de reţele de capitalizare a cunoştinţelor pentru proiectare.

Exemplu: sub-holonul proiectare produs poate exista într-o fabrică, proiectul poate fi transmis pe suport electronic altei fabrici, unde se va realiza prelucrarea propriu-zisă. Acolo i se asociază sub-holonul plan de prelucrare produs. Protocolul de schimb de informaţii trebuie să asigure propagarea modificărilor de proiectare astfel încât modificările la nivelul proiectului produsului să se regăsească la nivelul sub-holonului de plan procesare produs, iar acesta la rândul său să poată negocia cu holonul proiectare produs, o reproiectare din considerente de îmbunătăţire a fabricaţiei.

Prototipul de holarhie, datorită proprietăţilor de autonomie şi co-operare este proiectat să asigure funcţionarea şi evoluţia sistemului de fabricaţie. Distribuirea sub-holonilor specializaţi, în diferite locaţii spaţiale, poate fi asociată unui proces permanent de reconfigurare a sistemului, într-un sistem dinamic în care capacităţile de producţie suplimentare disponibile sunt într-o permanentă modificare.

2.2.2. MODEL DE SISTEM FRACTAL DE FABRICAŢIE

Termenul de fractal îşi are originea în geometria fractalilor, utilizată pentru a descrie şi analiza obiecte din spaţiul multidimensional, în mod deosebit focalizat pe dimensiunea fracţională unde geometria euclidiană nu este aplicabilă.

Definiţie. *Sistemul fractal de fabricaţie este o entitate-corporaţie acţionând independent, ale cărui obiective şi performanţe pot fi descrise precis [Warnecke,93].*

„Unitatea de fabricaţie”, ca descriere generică, este aceea care oferă servicii altor unităţi. Cu referire la teoria sistemelor, specific fractalilor este că relaţiile dintre ei sunt mai puternice în interiorul lor decât spre exterior.

Fractalul este un subset al Sistemului fractal de fabricaţie.

Caracteristicile fractalilor [Warnecke,93]:

a) sunt structuri auto-similare

- focalizare pe sistem local de obiective, ca subset al obiectivelor companiei;
- modularitate; sunt formaţi din subsisteme având elemente şi legături de acelaşi fel.

Proprietatea presupune recursivitate, paternuri în interiorul paternurilor. Seturile Mandelbrot din geometria fractală prezintă auto-similitudine deoarece ele nu numai că produc detalii la scară redusă, ci deasemenea produc detalii cu anumite măsuratori constante, deşi ele nu sunt identice.

Această proprietate se referă atât la caracteristici structurale ale proiectării organizaţionale, cât şi la modul în care fractalii realizează servicii sau formulează şi îndeplinesc obiectivele. Aceasta argumentează caracterul multidimensional al fractalilor şi susţine aserţiunea ***„fabrica fractală este o abordare integrată a domeniului fabricaţiei”*** [Warnecke,93]. Fractalii din sistemele de fabricaţie, ca şi cei din matematică, pot fi structuri similare, nu identice. Datorită varietăţii de soluţii posibile pentru probleme individule fractalii pot avea structuri interne diferite chiar dacă la nivel de bloc funcţional (I/O şi sistem de obiective) sunt identice.

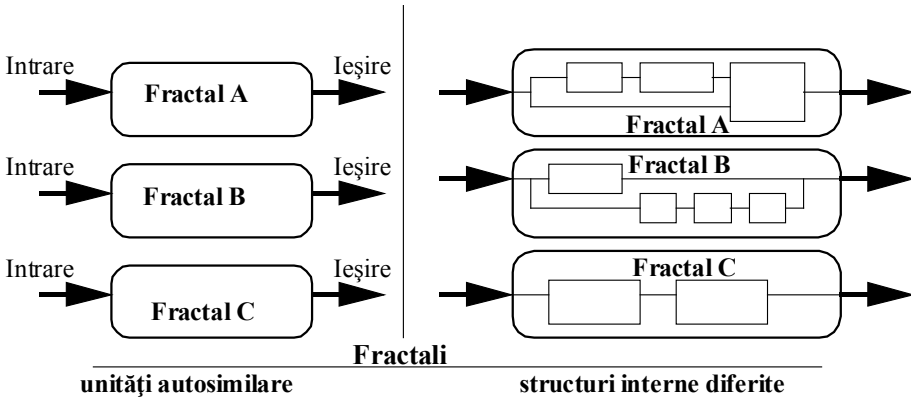


Fig. 2.6. Fractali autosimilari

Într-un sistem fractal de fabricaţie funcţiile centralizate sunt necesare şi se dezvoltă în corelaţie cu utilitatea lor în crearea lanţurilor de valoare adăugată. Se are în vedere planificarea resurselor la nivel central sau sprijinul oferit fractalilor pentru palnificare, concentrarea-capitalizarea cunoştinţelor în baze de date specializate, care nu pot fi menţinute la nivelul fiecărui fractal, dar care este disponibil fiecăruia dintre aceştia. Accesul fractalilor la bazele de cunoştinţe le dă posibilitatea de a-şi completa task-urile, păstrând un grad mare de autonomie.

Pentru task-urile care necesită resurse de care fractalul nu dispune se solicită serviciile altor fractali.

b) sunt structuri auto-organizate şi auto-optimizate

- ***operativ*** – procedurile sunt organizate optimal;
- ***tactic şi strategic***.

Fractalii îşi stabilesc structura proprie de obiective printr-un proces dinamic şi decid asupra contactelor interne şi externe. Fractalii se auto-crează, restructurează, regenerează şi dizolvă. Proprietatea implică toate nivelurile: operativ, tactic şi strategic, ceea ce face posibilă implementarea ideilor valoroase indiferent de la ce

nivel au sosit. Această abordare preia, în termeni specifici fractalilor, noţiunea japoneză de KAIZEN.

KAIZEN este procesul prin care se susţine, la toate nivelurile, efortul de a îmbunătăţi activitatea, constant şi gradat.
KAIZEN presupune eforturi pentru:

- ***„a face lucrurile mici, obişnuite mai bine " ; obiectivul este acela de a perfecţiona fiecare operatie, fiecare proces, indiferent cât de important este***
- ***„ a fi mai bun mâine decât azi" ; presupune perfecţionare continuă***
- ***„ stabilirea şi menţinerea continuă a unui standard ridicat"***
- ***„tratarea ficăruia ca pe un client."***

Masaaki Imai

c) sunt structuri orientate pe scop

Fractalii acţionează independent. Aceasta presupune că ei urmăresc îndeplinirea unui sistem propriu de obiective. Pentru ca ei să acţioneze ca un tot unitar este necesară menţinerea unei consistenţe a sistemului de scopuri ale tuturor fractalilor care interacţionează. Participarea şi coordonarea între fractali, sprijinită de un mecanism de moştenire asigură consistenţa sistemului de obiective elaborat. Prin ipoteză, obiectivele nu sunt impuse, ci se stabilesc prin acord mutual, în urma rezolvării conflictelor prin cooperare şi negociere, în cadrul unui proces iterativ, care conduce la noţiunea de **„împreună holostic"**.

d) sunt structuri dinamice

- structura organizatorică a fractalilor este capabilă de evoluţie/ autodezvoltare;
- adaptarea obiectivelor printr-o permanentă poziţionare şi evaluare a stării curente.

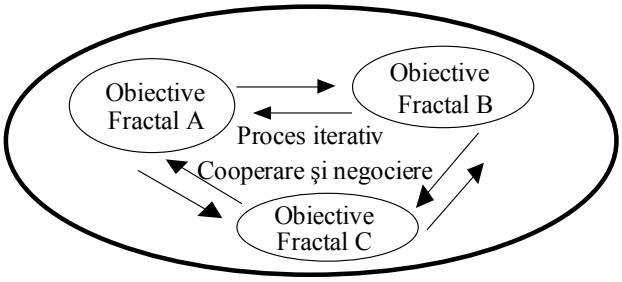


Fig. 2.7. Procesul de formare a sistemului de obiective

Fabrica fractală nu este prima abordare structurată a procesului de fabricaţie. Noţiunea tradiţională de segment de fabricaţie are o deficienţă majoră: natura sa statică.

segment de fabricaţie	*fractal*
• prelucrare • se structurează la momente specifice de timp • potrivit unui mediu stabil • activitate în conformitate cu sarcini specifice • sunt auto-responsabile • sunt evaluate pe baza rezultatelor	• asigură servicii – în sens larg • se structurează dynamic – proces constant de schimbare • potrivit pentru mediu schimbător • se integrează într-un proces de stabilire a obiectivelor • se auto-organizează şi auto-administrează • navigează

Tab. 2.1. Comparaţie dintre fractali şi segmente [Warnecke,93]

Fractalii trebuie să aibă caracteristicile de bază ale vitalităţii.

Definitie*. Vitalitatea este abilitatea de a acţiona cu succes în condiţiile influenţei unui mediu în schimbare.*

Termenul provine din biologie, având sensul de ,,forţa de a susţine viaţa". Aplicat la domeniul sistemelor inteligente, vitalitatea înseamnă caracteristica acestora de a se adapta şi a recţiona la influenţele mediului în care evoluează. Vitalitatea trebuie să înregistreze şi să evalueze în mod deosebit variabilele incluse în caracteristicile individuale [Sihn,95], care pot fi utilizate ca o măsură a schimbărilor caracteristicilor a şase niveluri denumite: **cultural, strategic, socio-psihologic, financiar, informaţional şi tehnologic**, interpretate ca dimensiuni ale unui mediu de lucru în care navighează fractalii.

Vitalitatea este influenţată de factori interni (*situaţia costurilor, potenţialul de fabricaţie, potenţial de cercetare şi dezvoltare, eficienţa managementului, potenţialul de livrare şi distribuţie, potenţialul financiar, logistica, amplasarea unităţii, resursa umană, domeniul de fabricaţie sau programul de produse*) şi de factori externi (*piaţa de desfacere, piaţa de furnizori, competiţie, legislaţie etc.).*
Interdependenţele şi influenţele dintre structurile organizaţiilor de tip fractal fac dificilă implementarea unui mediu CIM într-o fabrică fractală. Nevoia de autonomie, implicit inteligentă, a fractalilor, precum şi sistemul descentralizat de navigaţie trebuie să-şi găsească suportul informatic în CIM.

Navigaţia şi controlul fractalilor

Fractalii navighează în sensul că ei verifică permanent ,,poziţia" lor în zona de acţiune (traiectoria de stare), raportează nivelului central şi aplică corecţiile necesare.

În modelul fractal coordonarea centrală guvernează acţiunile unităţilor descentralizate. Direcţiile strategice şi managementul unei companii nu pot fi rezolvate prin şansă. Întreprinderea fractală trebuie să dispună de un sistem de management participativ, bazat pe delegare de responsabilitate, care necesită metode şi instrumente evoluate pentru modelare-simulare şi suport decizie. Prin combinarea metodologiei top-down şi bottom-up obiectivele strategice ale companiei fractale sunt elaborate central, iar printr-un proces iterativ de coordonare, colaborare şi negociere aceste macro-obiective generează obiectivele locale fiecărui fractal.

În fabrica fractală se susţine o metodologie botom-up pentru planificare şi control utilizând principiul „management prin excepţie", care asigură posibilitatea, la nivel central, de a preveni şi corecta eventualele erori.

Model de planificare producţie şi control de tip fractal

Un sistem fractal de planificare şi control a producţiei constă în fractali, ca unităţi de planificare independente.
Fractalul de planificare se compune din trei componente care implementează funcţii diferite:

- ***Componenta de planificare*** supervizează atât lanţul de aprovizionare din amonte, cât şi cerinţele posturilor din aval şi optimizează producţia dintr-o arie locală.
- ***Componenta de monitorizare şi control*** transferă rezultatele spre atelierele de fabricaţie şi capturează datele despre starea sistemului de producţie. Ea urmăreşte evoluţia procesului de fabricaţie şi compară starea curentă a sistemului cu starea planificată. Dacă sunt sesizate discrepanţe critice, componenta cere lansarea unui nou proces de optimizare a subsistemului fractal de fabricaţie.
- ***Componenta broker*** este responsabilă pentru implementarea funcţiei de comunicaţie şi negociere cu alţi fractali de panificare din întreprindere. Broker-ul primeşte toate informaţiile utile de la componentele de planificare şi control ale aceluiaşi fractal. El furnizează celorlalţi fractali de planificare noile informaţii de modificare a planului iniţial: reconfigurare capacitate de producţie, stocuri, cuante de timp etc. Deseori apar „conflicte" între fractalii de palnificare. Componentele Broker comunică, analizează situaţiile şi negociază soluţia acceptabilă pentru sistem.

Definiţie*. Lanţurile de alimentare reprezintă reţeaua de facilităţi, de la procurarea materiei prime la transformarea ei în semifabricate sau produse finale şi la distribuirea acestora .*

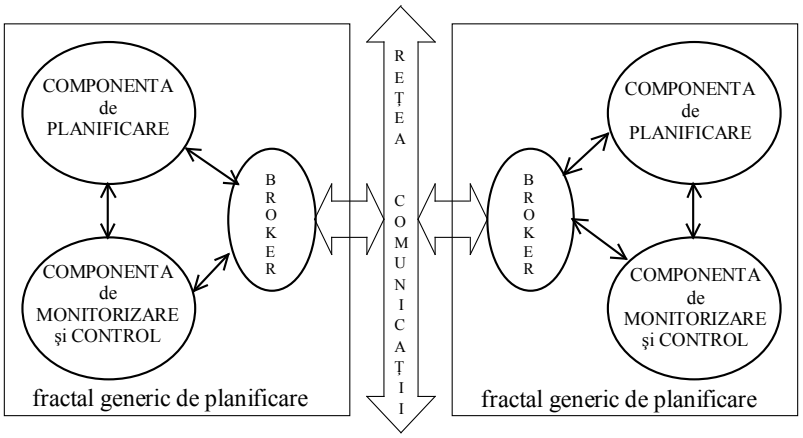


Fig. 2.8. Fractal generic de planificare şi control

În funcţie de structura internă a „*lanţurilor de alimentare*" (dependentă de gradul de complexitate şi de nivelul de incertitudine) se disting următoarele abordări:

- ***PP&C ierarhizat***

Fractalul PP&C include două unităţi de planificare, una pe termen scurt şi alta, grosieră, pe termen lung. Fractalul ierarhic superior realizează o planificare grosieră şi determină condiţiile de lansare şi încheiere a activităţii subordonaţilor. Fractalii subordonaţi execută planificările operaţionale preluând responsabilitatea îndeplinirii obligaţiilor atribuite lor. Nerespectarea planificării grosiere sau ignorarea, ei chiar, (fractalii au sisteme propri de obiective), se penalizează.
Performanţele planificării ierarhice depind de modul în care specificaţiile obţinute prin planificarea grosieră sunt transpuse în producţie. Dezavantajul major al abordării: lipsa reacţiei de la fractalul subordonat şi procesul de fabricaţie la fractalul superior ierarhic. Soluţia este aplicabilă pentru o producţie de nivel redus de complexitate şi grad mic de incertitudine, pentru care *prognoza* asupra producţiei este realizabilă.

- ***PP&C distribuit incluzând coordonarea***

Într-o arhitectură de sistem distribuit fractalul superior ierarhic preia funcţia de coordonator prin care propune fractalilor subordonaţi o anumită ordonare a activităţilor, având ca scop atingerea obiectivelor la nivel de sistem. Fractalii subordonaţi îşi vor dezvolta propriul sistem de obiective, pornind de la obiectivele globale de sistem.
Procesul de planificare se desfăşoară în următoarele faze:

1. *Fractalul coordonator realizează planificarea grosieră.* Planul de ordonanţare este transmis fractalilor subordonaţi.

2. *Fractalii subordonaţi execută planificarea pe termen scurt*, în funcţie de facilităţile de producţie de care dispun. Neîndeplinirea obligaţiilor asumate de un fractal poate genera conflicte cu ceilalţi fractali.
3. *Fractalii subordonaţi intraţi în conflict negociază* pentru rezolvarea acestora.
4. *Fractalii subordonaţi informează fractalul coordonator asupra planului de producţie*. Fractalul coordonator are acces la datele referitoare la „istoria" procesului.

– ***PP&C total distribuit***

Într-o arhitectură de sistem complet distribuit fractalii având aceleaşi nivel de autoritate, cooperează în scopul stabilirii unui plan global pentru reţeaua de producţie. Fractalii de planificare acţionează ca agenţi autonomi pe piaţă: cer cotaţii, fac oferte, dau comenzi (ordine), evaluează piaţa.

Un astfel de sistem reacţionează extrem de flexibil dar, datorită autonomiei fractalilor, procesul de comunicaţie şi negociere este adeseori prea lung.

– ***PP&C hibrid***

Pentru lanţuri de alimentare complexe, în urma unei analize a structurii entităţilor implicate şi a relaţiilor dintre ele, se pot formula soluţii specifice, hibride, combinând în mod convenabil avantajele fiecăreia din cele trei soluţii prezentate, având ca scop optimizarea comportării sistemul.

2.2.3. MODELUL DE SISTEM BIONIC DE FABRICAŢIE

Viaţa, ca structură şi organizare înseamnă autonomie, auto-organizare, comportare spontană, auto-refacere, creştere şi evoluţie având ca suport relaţii de tip ierarhic. Formele de viaţă sunt ordonate, începând cu structurile elementare, celulele, continuând cu organe, viaţa, şi la cel mai înalt nivel, societatea. Organele sunt formate din celule şi susţin, prin funcţiile pe care le îndeplinesc, forma de viaţă din care fac parte. **Sistemul bionic/biologic de fabricaţie** (SBF) transpune pentru fabricaţie trăsăturile viului şi stabileşte conceptele esenţiale şi metodologia realizării acestui tip de sistem [Oki,89, Oki,92, Oki,93, Ueda,92, Ueda,93]. în tabelul 2.2. se prezintă similitudinile/asocierile dintre lumea viului şi sistemele de fabricaţie.

Funcţionarea *celulei*, din punct de vedere biologic, este asigurată de *organite* specializate. Celulele sunt delimitate printr-o membrană, care permite schimbul de substanţe cu mediul chimic în care acestea există. Substanţele schimbate cu mediul asigură existenţa celulei şi îndeplinirea funcţiilor sale. Informaţiile schimbate de celule prin intermediul mediului sunt de natura „*flux chimic*". Celulele realizează multiple operaţii, ale căror implicaţii se răsfrâng chiar asupra celulei în sine. Coordonarea funcţională este asigurată de *enzime,* care au rol

de mediatori chimici între celule şi contribuie esenţial la menţinerea armoniei viului.

În fabricaţie, unităţile de producţie din *atelier,* celulele-flexibile de fabricaţie, sunt suţinute funcţional de echipamentele (organitele) care le compun. Fluxurile de materiale şi informaţii sunt cele care asigură integrarea unităţilor în structură, iar unităţile de coordonare (planificare şi control) au rolul de a asigura integrarea funcţională a unităţilor autonome.

Ca şi formele viului, unităţile din sistemul bionic de fabricaţie sunt ordonate ierarhic. Metodologia top-down este utilizată în procesul de stabilire a task-urilor specifice fiecărui nivel şi fiecărei unităţi în parte. Acţiunile cumulate ale unităţilor, printr-un proces bottom-up, se manifestă prin macro-operaţii la nivelul întregului sistem.

Funcţionarea în armonie a sistemului necesită coordonare.

Lumea viului	**Sistemele de fabricaţie**
• organitele	• echipamentele
• celulele	• celula flexibila de fabricaţie
• organele	• ateliere
• mediul chimic	• mediul – informaţii – materiale
• fluxul chimic	• fluxul de – informaţii – materiale
• enzimele	• coordonatori

Tab. 2.2. Similitudini între funcţionarea sistemelor biologice şi cele de fabricaţie

Celulele se multiplică prin diviziune repetată şi crează organe sau alte forme având funcţii specifice. Procesul de auto-creştere din lumea viului are la bază informaţia depozitată în ADN, care se regăseşte în fiecare celulă multiplicată. În sistemele de fabricaţie pot fi create modele pentru diverse unităţi funcţionale prin păstrarea/reutilizarea experienţei validate structurată în baze de cunoştinţe (tip informaţie genetică).

Pe baza similarităţilor dintre lumea viului şi sistemele de fabricaţie s-a propus termenul de **modelon,** ca element primar, care stă la baza creării SBF.
Modelonul este compus din [Oki,89] :

- ***modeloni de nivel inferior*** (ierarhie de modeloni-copii);
- ***operatori*** (acţiune enzimatică);

– ***memorie comună*** (schimb de informaţii între modeloni, istorie procese, baza de date şi cunoştinţe având rolul de *spaţiu de lucru*).

Modelonul, prin gradul său de generalitate, poate sta la baza realizării relaţiilor întreg-parte, a operaţiilor de auto-decizie, poate asigura integrarea şi armonia unităţilor autonome.

Ilustrarea unor potenţiale aplicaţii şi aspecte operaţionale din fabricaţia bionică sunt prezentate în [Oki,92], nefiind însă implementări pilot consistente de tipul celor din domeniul sistemelor holonice de fabricaţie.

2.2.4. COMPARAŢIE ÎNTRE MODELELE HOLONIC / FRACTAL / BIONIC

Conceptele descrise subliniază principiile de proiectare a sistemelor de fabricaţie flexibile ca structură şi operaţii. Acestea propun structuri bazate pe module autonome, distribuite, cooperative şi inteligente, holoni/fractali/celule, capabile să se auto-organizeze pentru a îndeplini funcţiunile necesare. Precizarea aspectelor parte/întreg, atât la nivel structural (organizarea holonilor, fractalilor, celulelor), cât şi funcţional (sitemul de obiective) se face printr-o procedură recursivă. Mecanismul creării holonilor se bazează pe descompunerea funcţională top-down a sistemului rezultând taskuri, completate prin asigurarea atributelor holonice (autonomie şi cooperare) şi a atributului inteligenţă.

Fractalii se definesc ca structuri puternic integrate în mediul în care evoluează, sunt dinamici, se auto-restructurează şi auto-reconfigurează ca reacţie la acesta.

În fabricaţia bionică diviziunea iniţială în celule limitează regruparea /reconfigurarea în crearea unei entităţi de fabricaţie.

Proiectarea unui sistem fractal de fabricaţie necesită o abordare multidimensională, tehnic/uman/cultural, în timp ce sistemul holonic de fabricaţie pune accentul pe o abordare orientată tehnic.

Această abordare a condus la obţinerea de soluţii reale privind implementarea SHF. Definirea *holonului NC*, *de control, de urmărire, diagnoză etc.* are ca scop transformarea în holoni (autonomi şi cooperativi), a unor funcţii specifice unui sistem evoluat de control automat a proceselor. Trăsăturile operaţionale ale unui sistem de fabricaţie evidenţiază legăturile funcţionale dintre unităţi. Aceste legături sunt asigurate prin fluxuri de informaţii (ierarhice şi heterarhice) având rolul menţinerii armoniei în operarea sistemului. Absenţa unui control centralizat (cel mai pregnant în paradigma fabricaţiei bionice) al unităţilor de comandă poate conduce la o comportare haotică a sistemului.

	Holonic	***Fractal***	***Bionic***
unităţi elementare	• ***holoni*** • *autonome şi cooperative* • *prin descompunere funcţională* • *unităţi predefinite* • *unităţi multidimensionale (inclusiv tehnic şi uman)*	• ***fractali*** • *autosimilări în obiective* • *sistem propriu de obiective* • *unităţi multidimensionale (tehnic,cultural,uman)*	• ***celule*** • *flexibilitate definită prin geneză* • *operaţii multiple*
grup	• ***holarhie*** • *set holoni cu funcţiuni specifice*	• *predefinie şi recursiv* • *regrupare dinamică* • *navigaţie*	• ***organe*** • *orientat spre nevoi funţionale* • *dinamice*
autonomie individuală	• *ridicată* • *cooperare şi negociere pentru stabilire obiective şi activităţi* • *limitată prin " canoni"*	• *ridicată* • *cooperare şi negociere* • *adaptabilitate prin vitalitate* • *bazată pe motivatie ptr.componenta antropocentrica*	• *ridicată* • *abilitatea viului de adaptabilitate* • *autodefinire operaţii*
Autonomia grupului	• *ridicată* • *cooperare şi negociere pentru stabilire obiective şi activităti* • *limitată prin " canoni"* • *forme intermediare stabile*	• *mostenită de la fractalii autosimilari* • *prin restructurare dinamică* • *prin auto-optimizare* • *atitudine antreprenorială*	• *funcţiuni predefinite ale oraganelor, prin geneză* • *autonomie operaţională ridicată*

Tab. 2.3. Modul de organizare în fabricaţia holonică, fractală şi bionică

În tabelul 2.4. coordonarea ierarhică precizează modul în care nivelurile adiacente sunt coordonate ca obiective şi activităţi. Comunicaţia dintre entităţile aflate pe acelaşi nivel poate fi denumită „comunicaţie laterală". Parametrii de planificare şi control arată modul în care entităţile îşi planifică şi controlează activităţile.

Aplicaţiile din SHF, rezolvarea unor probleme strict legate de interfaţă neutrală (driverul de dispozitiv), protocol de comunicaţii etc., crează cadrul de trecere de la definire concept la implementare, de la prezentare teoretică, la argumentare prin implementare tehnică, în mediu virtual sau în aplicaţii pilot industriale.

	Holonic	***Fractal***	***Bionic***
coordonae ierarhică	• *top-down pentru planuri* • *bottom-up decizii şi evaluare performanţe*	• *top-down şi bottom-up iterativ*	• *top-down specificaţii* • *bottom-up decizii*
comunicaţie laterală	• *comunicaţie şi cooperare între:* • *holonii din holarhie* • *între holarhii*	• *reţea de comunicaţii care să sustină:* • *cooperarea dîntre fractali* • *navigaţia*	• *indirectă prin:* • *mediu* • *enzime (coordonatori, catalizatori, operatori enzimatici)*
planificare şi control	• *planificare la nivel înalt (holoarhie)* • *dinamic şi concurent (heterarhic)* • *nevoie mare de comunicaţii*	• *proces continuu de modificare (adaptare) a obiectivelor* • *,,management prin exceptie"ca proces bottom-up* • *multiple bucle de control întreţesute* • *nevoie de comunicaţi (accent în interiorul fractalilor)* • *reacţie rapidă prin descentralizarea competenţelor în adoptarea deciziei*	• *minim* • *reacţii la situaţii de urgenţă* • *într-un proces concurent* • *total descentralizat*

Tab.2.4. Coordonarea ierarhică

Multimodelarea se constituie în suportul necesar acoperirii întregului ,,ciclu de viaţă" al unui sistem evoluat de fabricaţie.

Introducerea holonului personal aduce în universul holonilor clasă personal, componenta antropocentrică, dar mai puţin precizată decât celelalte clase de holoni. Mediul de testare pentru explorarea socio-tehnică nu este încă realizat, în comparaţie cu mediul de testare a sistemului tehnologic holon de fabricaţie.

Proiectarea holarhiilor face apel la metodologie orientată obiect şi la abordare structurată. Metodele utilizate pentru fractali propun o analiză multi-dimensională. Lipsa unor descrieri cantitative pentru metodele utilizate în fabrica fractal fac dificilă implementarea acesteia.

Sistemul bionic de fabricaţie se realizează prin geneză, care dispune de flexibilitatea de a crea forme diferite în concordanţă cu nevoile sistemului. El are

funcţii care modelează diviziunea celulară (multiplinarea unităţilor de bază), precizează operatorii enzimatici şi functiile unităţilor de bază (celule).

2.3. FLEXIBILITATEA SISTEMELOR DE FABRICAŢIE

Indiferent dacă este vorba de CIM sau universul post CIM cerinţa de flexibilitate a sistemului de fabricaţie este una din condiţiile de creştere a competitivităţii acestuia. Flexibilitatea este un factor major în automatizarea producţiei. Deşi conceptul de flexibilitate a sistemelor de fabricaţie este foarte important, nu există încă o definiţie acceptată internaţional însă diferitele definiţii existente utilizează în comun aceleaşi concepte generale, aşa cum rezultă şi din:

Definiţia 1*: *Un sistem flexibil de fabricaţie *(SFF) este un aranjament de maşini (centre de prelucrare cu control numeric, dotate cu dispozitive automate de schimbare a sculelor) interconectate prin intermediul unui sitem de transport. Sistemul de transport furnizează piesele pentru maşinile de prelucrare pe palete sau alte unităţi de interfaţă, astfel încât deservirea maşinilor să fie rapidă, precisă şi automată. Un calculator central conduce maşinile şi sistemul de transport. SFF poate procesa simultan diferite tipuri de piese. (National Institute of Standards and Technology-NIST).*

Definiţia 2*: *Un SFF *este o unitate de producţie capabilă să realizeze o gamă de piese discrete cu un minimum de intervenţie umană. El constă din staţii de lucru-maşini unelte, echipamente de fabricaţie, asamblare sau tratament) interconectate printr-un sistem de manipulare a materialelor destinat deplasării pieselor de la o staţie de lucru la alta şi operează ca un sistem integrat sub conducere complet programabilă (United States Office of Technology Assessment).*

Definiţia 3*: *Un SFF *este un sistem de fabricaţie automat, capabil să producă cu un minimum de intervenţie umană, produse dintr-o gamă dată. Flexibilitatea sistemului va fi de regulă restrânsă la o familie de produse, pentru care acesta a fost proiectat. (International Institution for Production Engineering Research-CIRP).*

Definiţia 4*: *Un SFF *este un complex integrat condus de calculator, alcătuit din maşini-unelte cu comandă numerică, dispozitive automate de manipulare a materialelor şi a sculelor şi echipamente automate de masură şi testare care, cu un minimum de intervenţie umană şi timp de modificare scurt, poate să proceseze*

produse aparţinând unei anumite familii specifice de produse, conform unei planificări anterioare (Economic Commission for Europe).

Definiţia 5*: *Un sitem flexibil de fabricaţie *este acela care are capacitatea de a se adapta la o gamă largă de situaţii specifice fabricaţiei. El este o soluţie având abilitatea de a supravieţui în confruntarea cu o cantitate considerabilă de constrângeri care, principial sunt absente, în problema originală.*

Flexibilitatea este o problemă relativă depinzând de domeniul situaţiilor posibile din mediu. Mediul este un set al seturilor de constrângeri, soluţiile flexibile trebuind să supravieţuiască în confruntarea cu oricare dintre elementele acestuia.

Când se compară SFF operaţionale în industrie este firesc să se ţină cont că ele diferă prin numărul de maşini controlate în sistem, costurile de investiţii etc., ceea ce face necesară stabilirea următoarelor destincţii între dimensiunile lor :

- ***UFF (Unitate Flexibilă de Fabricaţie)***: este uzual un centru de prelucrare echipat cu o magazie multipaletă, un schimbător automat de palete sau un robot şi un dispozitiv de schimbat automat scula (unitatea este capabilă să lucreze parţial autonom, fără supraveghetor uman);
- ***CFF (Celula Flexibila de Fabricaţie)*** : este compusă din mai mult de două maşini, din care cel puţin o maşina este centru de prelucrare, magazii multipaleta, încărcător automat de palete şi schimbătoare de scule pentru fiecare maşină. Maşinile, la fel ca şi operaţiile efectuate de celulă sunt controlate de către un calculator central;
- ***SFF (Sistem Flexibil de Fabricaţie)*** : este constituit din două sau mai multe UFF sau CFF conectate printr-un sistem automat de transport, vehicule autonome ghidate, elemente de încărcare/descărcare controlate de computer, benzi transportoare care deplasează palete cu piese, semi-fabricate, produse finite sau scule, între maşini şi magazii. Întregul sistem este controlat de un computer, care este conectat la sistemul informatic al fabricii.

Dintre elementele comune, definitorii ale SFF se pot preciza:
- conducerea de către calculator;
- integrarea sistemului din punct de vedere al conducerii;
- sistem logistic, ca resursă de sistem partajată, care asigură interconectarea subansamblelor din sistem;
- intervenţia minimă a operatorului uman;
- capacitatea de a realiza produse dintr-o familie specifică, pe baza unei planificări.

Criterii de flexibilitate în sistemele de fabricaţie

Flexibilitatea obţinută prin automatizare este principala caracteristica a SFF. În acest context flexibilitatea este gândită uzual ca uşurinţa cu care sistemul poate fi iniţializat pentru a procesa o varietate de piese. Acesta este însă numai un criteriu de flexibilitate, adesea denumit ***flexibilitatea produsului***.

Pentru a evalua un SFF există un număr de alte criterii de flexibilitate care trebuie luate în calcul :

- ***flexibilitatea maşinii:*** flexibilitatea cu care maşinile din sistem pot fi iniţializate referitor la utilizarea dispozitivelor, a elementelor de fixare şi poziţionare, a programelor pentru comenzile numerice, pentru a procesa repere dintr-o familie dată;
- ***flexibilitatea procesului***: posibilitatea de a produce un set de tipuri de repere, în diferite moduri, utilizând diferite materiale;
- ***flexibilitatea produsului***: abilitatea de a trece la producţia unei noi familii de repere în mod economic şi rapid;
- ***flexibilitatea de rutare:*** posibilitatea sistemului de a continua operarea prin alegerea de căi alternative pentru piese în situaţia defectării unor resurse de sistem. Implică faptul că funcţiile maşinilor defectate pot fi preluate de către alte maşini din sistem;
- ***flexibilitatea volumului:*** posibilitatea SFF de a lucra eficient la diferite volume ale producţiei;
- ***flexibilitatea extinderii:*** capacitatea de a construi sistemul şi de a-l extinde în funcţie de necesităţi , uşor şi modular;
- ***flexibilitatea de operare:*** abilitatea de a modifica ordinea unor operaţii asociate fabricaţiei unor repere.

Relativ la flexibilitatea sistemului de producţie se mai pot preciza:

- ***flexibilitate improprie****:* obţinută prin creşterea capacităţii de producţie prin duplicare echipamente, creşterea vitezei de prelucrare etc. Se consideră că simplitatea metodei este un avantaj, metoda fiind totuşi extensivă;
- ***flexibilitatea prin configurabilitate:*** este legată de posibilitatea de a configura şi reconfigura rapid capacităţile de producţie, astfel încât gama produselor să poată varia foarte mult;
- ***flexibilitatea prin calibrare;*** calibrarea este un proces complicat, având însă avantajul că asigură utilizarea optimă a resurselor de fabricaţie.

CONCLUZII

Capitolul 2 îşi propune stabilirea cerinţelor pentru sistemul viitor de fabricaţie care, indiferent că termenul utilizat este ***NGMS*** *(Next Generation Manufacturing Systems),* ***ADMS*** *(Advance Manufacturing Systems),* ***DEE*** *(Dinamic Extended Enterprise),* ***IMS****(Intelligent Manufacturing Systems) sau* ***CE*** *(Concurrent Enterprise), având sprijinul oferit de noile modele de referinţă ale fabricii (holonic, fractal, bionic, virtual), va fi rezultatul presiunii exercitate de dezvoltarea tehnologiilor informatice şi de comunicaţii în medii electronice virtuale şi va avea ca elemente caracteristice* ***autonomia****,* ***co-operarea****,* ***interdisciplinaritatea****,* ***abordarea integrată produs-proces-sistem****,* ***reconsiderarea resursei umane****, va fi* ***agilă în organizare*** *şi* ***virtuală ca asociere şi distribuţie geografică****.*

Structurând cerinţele pentru noul sistem de fabricaţie, corespunzător celor cinci paradigme care caracterizează un sistem de fabricaţie: paradigma managementului financiar, paradigma organizării, paradigma controlului, paradigma tehnologică şi paradigma proiectării au rezultat următoarele:

1. Noul cadru, exterior companiei, impune înlocuirea *,,integrării pe verticală a întreprinderii ⇔ piramida oragnizatorică****”,*** ***bazată pe tehnologii informatice şi structuri ierarhice de control aferente CIM prin*** *,,forme organizatorice elastice, de tip colaborativ, caracterizate prin autonomie şi coordonare multi-agent ⇔ casa organizatorică”.*

2. Nouă întreprindere va avea *funcţii care imită organismele vii* ***(auto-organizare, auto-refacere, creştere şi evoluţie), va*** *utiliza modele şi va propune metodologii care să acceseze toate nivelurile procesului de afaceri****, va răspunde rapid-orientat client la cerinţele pieţii,*** *se va armoniza cu mediul* ***(va răspunde cerinţelor ecologice) şi*** *va deţine informaţii sistematizate relativ la întregul ciclu de viaţă al produsului obţinut în întreprinderea virtuală.*

3. Paradigma proiectării propune *metodologia interdisciplinară de proiectare* ***prin utilizarea ingineriei concurente în locul celei seriale.***

4. Paradigma proiectării noii întreprinderi se bazează pe modele de proiectare iterativă, care să asigure realizarea sistemului dorit prin modelare, prototipizare rapidă şi simulare, considerând ca referinţă modelul spiral al ciclului de viaţă.

În cadrul acestui capitol se face o analiză a ***vectorilor de schimbare*** *:* ***modelul de sistem holonic*** *de fabricaţie,* ***modelul de sistem fractal*** *de fabricaţie,* ***modelul de sistem bionic*** *de fabricaţie. Pentru fiecare din această nouă paradigmă din universul modelelor de sisteme de fabricaţie se prezintă elementete specifice, iar în final se face o analiză comparativă.*

Din această analiză rezultă că noile paradigme propun structuri bazate pe module autonome, distribuite, cooperative şi inteligente, holoni/fractali/celule, capabile să se auto-organizeze pentru a îndeplini funcţiunile necesare, iar precizarea aspectelor parte/întreg, atât la nivel structural (organizarea holonilor, fractalilor, celulelor), cât şi funcţional (sistemul de obiective) se face printr-o procedură recursivă, dar sunt încă în faza de cadru conceptual, nesemnalându-se realizari în implementare similare sau măcar apropiate de cele obţinute prin metodologiile dezvoltate în cadrul paradigmei CIM.

Se subliniază că, dintre toate aceste noi paradigme, cea care este cea mai ***aproape*** *de a propune o soluţie de implementare, pentru un sistem de fabricaţie, este modelul holonic. Aplicaţiile dezvoltate pe baza conceptului de holon şi sistem holonic-holarhie, rezolvarea unor probleme strict legate de interfaţă neutrală (driverul de dispozitiv), protocol de comunicaţii etc., crează cadrul de trecere de la definire concept la implementare, de la prezentare teoretică la argumentare prin implementare tehnică, în mediu virtual de simulare sau în aplicaţii pilot industriale.*

Se subliniază nevoia unui cadru de multimodelare care să susţină eforturile de trecere de la propunere conceptuală la simulare şi apoi la tehnici de implementare aplicaţii reale.

Capitolul 3 – cuprins

3.1. Modelul fabricii virtuale **73**

3.1.1. Arhitectura de fabrică virtuală **75**

3.1.2. Obiecte, procese, agenţi în universul fabricii virtuale **78**

3.1.3. Problema alocării activităţilor **80**

3.1.4. Definirea noţiunii de fabricator **83**

3.2. Platforma multi – dimensională pentru modelare **85**

3.2.1. Dimensiunea – reprezentare funcţională **88**

3.2.2. Dimensiunea – reprezentare structurală **93**

3.2.3. Dimensiunea – reprezentare comportamentală **95**

- **Prezentare conceptuală a Sistemelor cu Evenimente Discrete (SED)** **96**
- **Caracteristicile modelului SED pentru sistemele de fabricaţie** **98**
- **Modelarea SED cu reţele Petri** **98**

3.2.4. Dimensiunea - reprezentare contextuală **104**

Concluzii **109**

3

FABRICA VIRTUALĂ

3.1. MODELUL FABRICII VIRTUALE

Fabrica virtuală (**VF**) se constituie într-un cadru de multimodelare care să asigure:

- ***implementarea controlului adaptiv***, cu referire la incertitudinile mediului de fabricaţie datorate factorilor externi impredictibili (uzură şi defecte, comenzi eronate ale operatorului uman, schimbarea opţiunii clienţilor, [Lin,93];
- ***coordonarea agenţilor autonomi***; operatorii umani şi maşinile cooperează în vederea atingerii structurii scop [Levis,94];
- ***controlul inteligent*** bazat pe tehnici neuro-fuzzy şi algoritmi genetici pentru a realiza activităţi complexe de control evoluat [Dumitrache,93] prin proiectare, modelare, simulare şi prototipizare în mediu virtual a unor sisteme de fabricaţie evoluate.

Fabrica virtuală asigură platforma de integrare a controlului avansat cu funcţiunile de afaceri ale sistemului de fabricaţie [Raulefs,94], şi oferă cadrul necesar dezvoltării întreprinderii, pe baza unui model cu învăţare, în conformitate cu o strategie de dezvoltare incrementală. Abordarea orientată pe platforma de multimodelare pentru: suport decizie, dezvoltare şi operare sistem de fabricaţie manifestă o tendinţă convergentă cu topica curentă de cercetare în direcţia Integrării în Fabricaţie (IiM). În domeniul proiectării şi operării sistemelor de control automat se acordă o atenţie deosebită facilităţilor oferite de mediile virtuale pentru modelare, simulare şi validare algoritmi, tehnici şi soluţii de sistem de automatizare, bazate pe calculator, pentru fabricaţie avansată.

Ţinând cont că FMS (*Flaxible Manufacturing Systems*)- Sistemele Flexibile de Fabricaţie, sunt încă departe de a-şi atinge potenţialul cerut de fabrica viitorului, *realitatea arată că la nivelul sistemului de fabricaţie propriu-zis (secţie, atelier, echipament) sistemele de control cu calculator nu sunt atât de flexibile şi integrate cum se precizează în lucrările academice [Rembold,93],* modelarea şi simularea, ca tehnici de dezvoltare rapidă de produse, procese şi sisteme, la nivelul fabricii

virtuale, se constituie într-o abordare de perspectivă în creşterea calităţii proiectării prin utilizarea ingineriei concurente şi utilizarea mediilor virtuale susţinute de dezvoltarea suportului informatic şi a mediilor de comunicaţii.

Sistemele de control la nivelul secţiei sunt focalizate pe informaţii, colecţii de date şi monitorizare mai mult decât pe controlul sistemului complex, distribuit. Multe FMS-uri sunt formate prin interconectarea unor structuri puternic orientate hard, nestandardizate, greu de interconectat şi care nu oferă accesul la informaţiile de control şi stare, ceea ce determină o slabă utilizare a resurselor oferite de calculator pentru controlul avansat [Aldemo,95]. Utilizatorii de FMS încearcă să standardizeze produsele şi să reducă diversitatea loturilor pentru a uşura operarea sistemelor şi a justifica economic utilizarea SFCS (*Shop Floor Control Systems*).

Termenul de ***Fabrică Virtuală*** a fost utilizat iniţial pentru a identifica o ***"fabrică agilă",*** dar conceptul face referire la un mediu virtual în care sistemele software pentru controlul cu reacţie al proceselor de fabricaţie şi diferite scenarii de acţiune pot fi testate şi executate fără a utiliza direct echipamentele fizice. Modulele software de control interacţionează cu o suită de module software care simulează atât funcţionarea echipamentelor, cât şi procesele de fabricaţie.

Dacă în programele uzuale de simulare fabricaţie, facilităţile pot fi definite prin parametrii (vezi SIMAN-CINEMA), conceptul de fabrică virtuală face apel în mod deosebit la modelul virtua „real" al facilităţilor, la modele ale facilităţilor atât din punct de vedere al unor caracteristici privind funcţiile îndeplinite, pentru a putea fi integrate în SFF, cât şi din punct de vedere al propriilor performanţe, legate de funcţionare (dimensiuni reale, viteze reale, încărcări reale etc.) care să permită simularea proceselor complexe de fabricaţie în mediu virtual şi posibilitatea ulterioară de „încărcare a aplicaţiei direct în mediul real", pe maşina virtuală „identică" cu cea din mediul simulat [Bodner,97].

Programarea orientată pe obiecte (OOP) şi modelarea solidelor din ingineria mecanică au deschis calea pentru realizarea de astfel de aplicaţii complexe de simulare. Tehnologia de interfaţare cu realitatea virtuală şi animaţia 3D pot crea iluzia unei fabricaţii reale, interactive. Sunt anunţate primele realizări de ***„full virtual prototyping"*** care presupun prototipizare virtuală, simulare prin tehnologia de interfaţare cu realitatea virtuală (etapa care implică participarea activă a beneficiarilor), urmată de prototipizarea rapidă a modelului. „Exerciţiul" trece dincolo de experiment fiind obţinute deja produse industriale reale, fabricabile, utilizabile (de exemplu: telefon mobil [Tuikka,97]).

Problematica fabricii virtuale este legată de ***modelare-simulare-validare în*** *mediu virtual* pentru:

- *proiectarea unei noi fabrici;*
- *instalarea unei noi linii de producţie sau a unui nou echipament;*
- *introducerea unui nou produs;*
- *instalarea unei noi tehnologii;*
- *schimbarea strategiei de control utilizate într-un sistem;*
- *managementul de resurse.*

Scopul major al VF este **experimentul în mediu simulat** prin:

- *reprezentarea detaliată a sistemului de fabricaţie în software;*
- *dezvoltarea unei discipline inginereşti pentru modelare ca suport în proiectare;*
- *integrare pachete software şi metodologii pentru rezolvarea de probleme complexe;*
- *autoanalizarea procesului de modelare ca suport proiectare.*

VF utilizează seturi integrate de pachete software pentru proiectare, optimizare, experimentare, modelare, iar eforturile sunt orientate către:

- *reprezentarea datelor ca suport pentru integrarea metodologiilor şi pachetelor software (simulare, optimizare etc.);*
- *utilizarea de modele de referinţă pentru sistemul de fabricaţie, care să sprijine procesul de modelare şi experimentul virtual;*
- *sinergia, integrarea şi interoperabilitatea programelor software.*

Definţie: ***Fabrica virtuală este un sistem de modele şi aplicaţii software*** *care descriu în mediu virtual evoluţia dinamică a unui sistem de fabricaţie real, cu scopul rezolvării următoarelor clase de probleme:*

- *asistarea adoptării deciziei strategice/tactice/operaţionale;*
- *dezvoltarea de produs /procese /facilităţi de fabricaţie;*
- *dezvoltarea sistemului automat de control;*
- *managementul de procese;*
- *alocarea resurselor.*

Contextul integrării fabricii virtuale în universul de afaceri este prezentat în figurii 3.1.

3.1.1. ARHITECTURA DE FABRICĂ VIRTUALĂ

Se consideră cadrul de modelare al fabricii virtuale constând din patru subsisteme:

- ***subsistemul*** *management afaceri;*
- ***subsistemul*** *modelare produs /proces;*

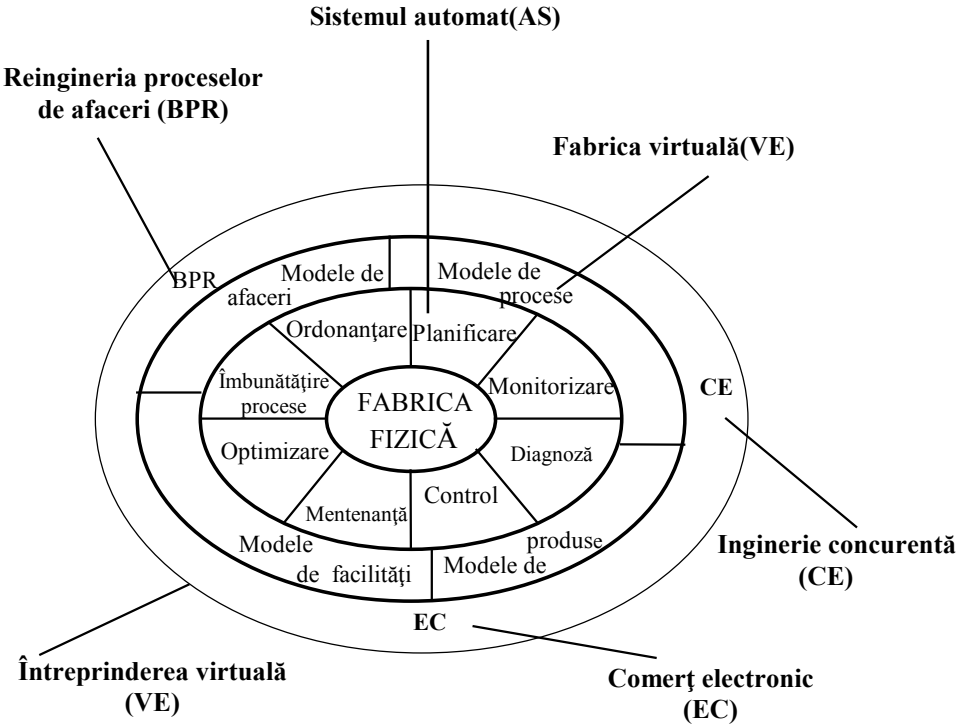


Fig. 3.1. Contextul dezvoltarii conceptului de fabrică virtuală

- subsistemul **facilităţi** cuprind entităţile fizice ale sistemului de fabricaţie;
- subsistemul **management de procese**.

Sistemul de management procese iniţiază crearea structurii scop a fabricatorului în corelaţie cu funcţii de afaceri (comenzi, costuri, beneficii etc.) şi modelele de produse. Mulţimea proceselor alimentează agenţii cu procesele cerute pentru satisfacerea structurii de scop.

Subsistemul de automatizare, AS este mediul de execuţie pentru fabrica virtuală, asigurând interacţiunea dintre „universul modelelor” şi “universul fizic”, al sistemului de fabricaţie propriu-zis, pe care primul îl modelează. O cerinţă majoră pentru subsistemul de automatizare este aceea de a avea o structură consistentă atât cu modele de calcul distribuit, cât şi cu produse din domeniul comercial deoarece ambele evoluează rapid, iar sistemul de control automat trebuie să asigure independenţa şi consistenţa (izolarea) fabricii virtuale faţă de modificările de la nivelul fizic al echipamentelor.

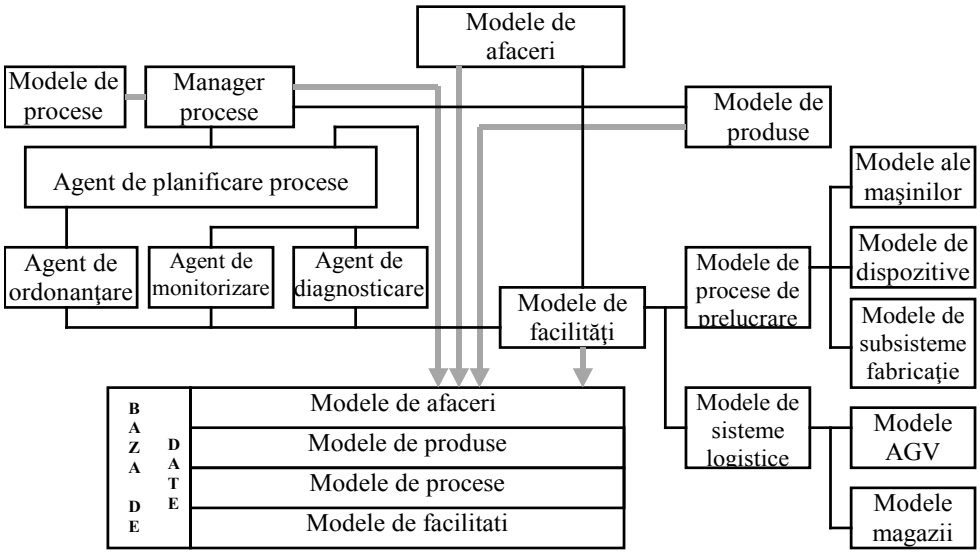


Fig.3.2. Fabrica Virtuală

În practica sistemelor de control al fabricaţiei, controlul atelierului este orientat în mod deosebit pe operaţiile de nivel înalt. Schimbarea acestuia este dificilă, cu implicaţii financiare şi efort de proiectare deoarece majoritatea echipamentelor de control sunt echipamente dedicate. Această situaţie se constituie într-o dificultate majoră în integrarea echipamentelor de automatizare, în monitorizarea stării echipamentelor, pentru controlul datelor de producţie şi introducerea de noi tehnici de ordonanţare pe baza controlului supervizor al procesului de fabricaţie.

Este unanim acceptată aserţiunea ***„resursele de fabricaţie acţionează ca agenţi care generează procese având ca obiectiv satisfacerea unei structuri scop”*** furnizată de „motorul” de decizie orientat de reguli de afaceri, pentru a asigura livrarea produselor în conformitate cu un model de date produs (PDM- *Product Data Model*) dezvoltat în mediul de inginerie concurentă (CE-*Concurrent Engineering*) Conceptele cheie utilizate sunt: *structura scop, procese de fabricaţie, structura de procese, situaţii-operaţii, protocol de comunicaţii între scena externă şi cea intern , agenţi autonomi, istoria unui agent*.

„Jucătorii cheie”, *proiectantul, administratorul afacerii, managerul de procese* ***cooperează*** pentru a asigura funcţionalitatea fabricii programabile. Macro-agenţii care realizează controlul şi ordonantarea sunt: ***Agentul de Planificare Procese***, ***Agentul de Diagnostic Procese***, ***Agentul de Monitorizare Procese***.

3.1.2. OBIECTE, PROCESE, AGENŢI ÎN UNIVERSUL FABRICII VIRTUALE

Pentru a modela fabrica virtuală se propun tehnici de programare orientate pe obiecte (OO), cu următoarele definiţii [Raulefs,94]:

Context = Set Obiecte
Obiect = Id x Stare x Set Operaţii
Stare = Variabilă → Valoare
Operaţie = (Stare x Mesaj x Context) x (Stare x Set Mesaje x Context)
Mesaj = ListaValori
Comportare = (Mesaj x Context) x (Set Mesaje x Context)

Contextul este creat de setul de obiecte asociate. Obiectele comunică prin mesaje care pot determina schimbarea stării, generarea de mesaje către alte obiecte, crearea sau ştergerea unor obiecte din Context.

Comportarea unui obiect stabileşte modul în care obiectul răspunde la un mesaj (din setul de mesaje acceptat) şi este determinat de starea în care se găseşte obiectul şi operaţiile aferente.

Procesele se definesc ca activităţi extinse temporal, realizate de agenţi şi obiecte. Situaţiile şi operaţiile sunt elemente primare care descriu procesele.

Situaţia constă într-o aserţiune şi o constrângere de interval de timp, Situaţie: = [Aserţiune/constrângere de timp].

Constrângerea de timp este mulţimea intervalelor pentru care aserţiunea este valabilă.

Evenimentul semnifică o schimbare care are loc la un anumit moment de timp.

Operaţia descrie un mecanism corelat cu un eveniment. O operaţie poate fi considerată o instanţă reprezentând un set de evenimente care diferă prin proprietăţi determinate de context (de exemplu: valoarea intrărilor) şi prin rezultatele (ieşirile) obţinute în urma realizării acestora.

Executarea unui proces corespunde unei reţele de situaţii-operaţii asociate unei reţele de situaţii-evenimente.
Reţeaua de situaţii-evenimente constituie ***istoria procesului***.

Scena de intrare este stabilită de condiţiile externe care induc o operaţie asupra unui agent sau obiect şi este determinată de scenele de ieşire ale proceselor care interacţionează cu procesul dat.

Protocolul de interacţiune dintre două procese este relaţia dintre scenele de ieşire şi cele de intrare a proceselor.

Structura scop se referă la ce intenţionează să realizeze un sistem.

Specificarea cerinţelor pentru proiectarea sistemului de fabricaţie integrat se poate face în două faze independente:

- ***modelarea conceptuală*** a proceselor de afaceri şi inginerie a afacerii pentru cerinţe preliminare la nivel de întreprindere, prezentate într-un limbaj de modelare intermediar, bazat pe un anumit meta-model;
- ***specificare formală*** prin care modelul conceptual este rafinat şi transformat într-un sistem de construcţii formale pentru care se pot stabili reguli formale, se poate proiecta prototipul etc.

Faza de modelare conceptuală este un proces în care toate cunoştinţele achiziţionate trebuie structurate în termeni ai conceptelor unui meta-model ales (care balansează între modelarea dependenţelor de timp şi semantica fluxurilor), de exemplu: modelul care poate complementa modelul ISO/IEC,91, [Dardenne,91] ***scenariu-jucător-episod-eveniment***.

Strategia de achiziţie a cunoştinţelor este orientată-agent, ceea ce corespunde cerinţei de modelare de sisteme existente (cu agenţi şi structură organizatorică pre-stabilite).

Metodologia propusă are la bază patru etape:

a) Identificarea agenţilor şi structurii lor

Se identifică procesele la nivelul cel mai de sus al agenţilor urmată de descompunerea în subagenţi, ca proces iterativ, până se obţine un nivel stabilit de descentralizare în care se pot identifica atribute sau relaţii ce pot fi asignate unui singur agent;

b) Identificaea scopurilor şi a constrângerilor

Elaboratea structurii scop constă în identificarea scopului pentru fiecare agent şi crearea lanţului de moştenire părinte-copii pentru structura de scopuri, meta-relaţia „reducere” pentru capturarea structurii scop-subscop, respectiv definirea unor instanţe, meta-relaţii între agenţi şi scopurile lor, de tipul „este_responsabil”. Scopul final asociat unui agent este ***„scop ţintă”*** (scop „frunză”). *Operaţionalizarea scopurilor* se face prin translatare în *constrângeri.*

c) ***Definirea scenariilor***

Agenţii efectuează acţiuni astfel încât constrângerile să satisfacă structura de scopuri. Se urmăresc *scenarii*, care în forma conceptuală au la bază un formalism grafic, în termeni de relaţii cauzale între evenimente şi acţiuni. Pentru a distinge acţiuni care implică un singur agent, de acţiuni care implică mai mulţi agenţi se introduce noţiunea de *interacţiune*, acţiunea de schimb de mesaje între agenţi.

Rolul unui agent este mulţimea de acţiuni şi interacţiuni stabilite de agent pentru a-şi satisface cerinţa scop.

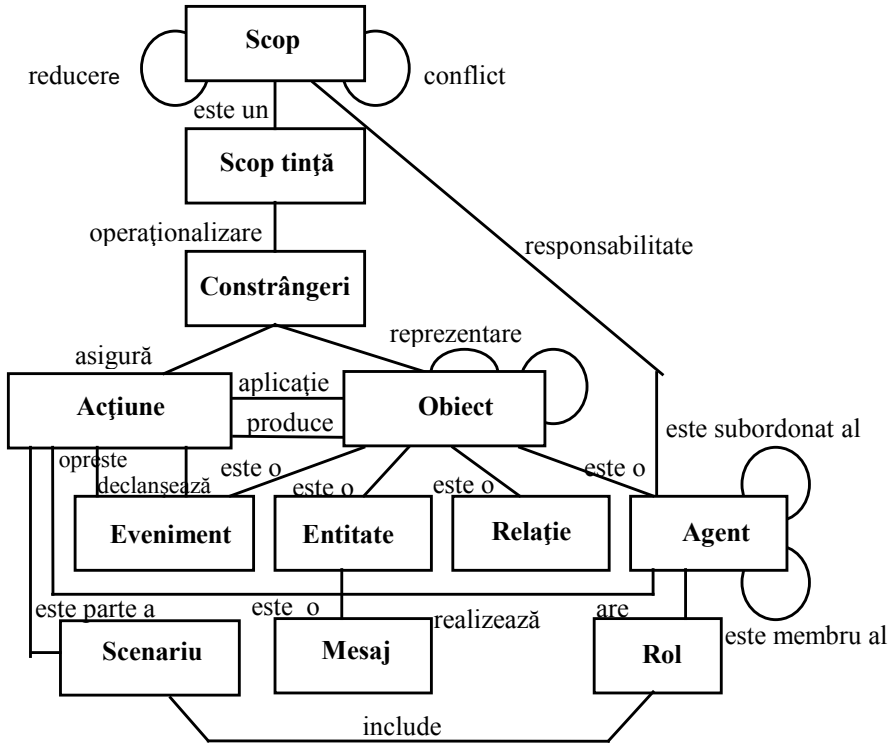


Fig. 3.3. Model meta-conceptual

d) Caracterizarea obiectelor/acţiunilor

Meta-structurile care caracterizează complet, în termeni de atribute şi legături, obiectele şi acţiunile care apar în specificaţii includ: durata, condiţii pre şi post-declanşare, condiţii de stop pentru acţiuni şi stări pentru obiecte.

3.1.3. PROBLEMA ALOCĂRII ACTIVITĂŢILOR

Descompunerea unei funcţii în sub-funcţii şi alocarea acestora unor agenţi este o procedură cunoscută. Atunci când agenţii sunt oameni sau maşini inteligente, fiecare agent contribuie la executarea mai multor funcţii, secvenţial sau concurent, ceea ce face ca alocarea funcţie – agent să nu fie una clasică. Alocarea trebuie facută astfel încât sarcina rezultată pentru fiecare agent să nu depăşească capacitatea acestuia.

Modelarea alocării funcţiilor ţine cont de încărcarea fiecărui agent precum şi de efortul necesar coordonării interacţiunii dintre aceştia.

Primul pas în rezolvarea problemei alocării activităţilor este cea a descompunerii funcţionale. Se utilizează descompunerea în ***scopuri – obiective – activităţi*** (frecvent utilizată când se face referire la agenţi umani) sau în termeni de ***misiuni – funcţii – activităţi.***

În figura 3.4. se propune o descompunere funcţională întreţesută, corespunzătoare, structurată pe nivelurile arhitecturii ierarhice de control (atelier, celulă, staţie de lucru, echipament).

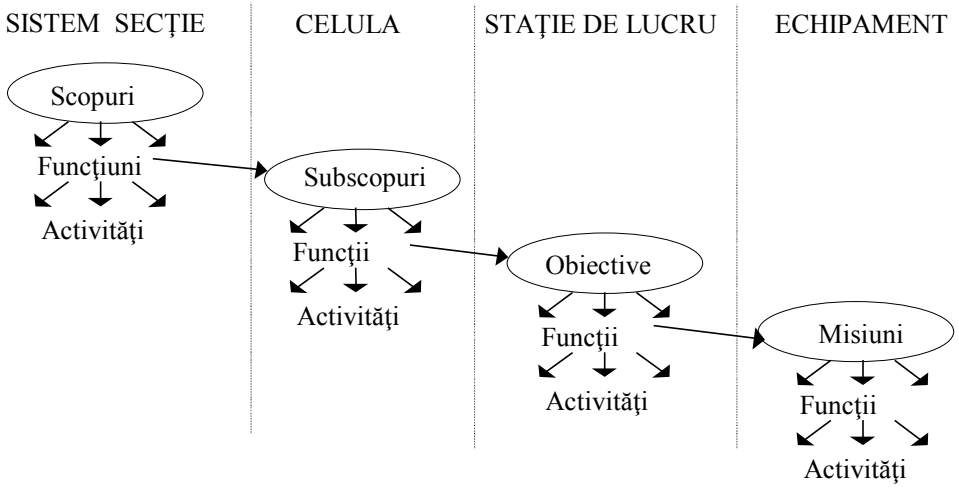


Fig. 3.4. Decompoziţie întreţesută pentru funcţiile de sistem

Descompunerea funcţională se poate face la ficare nivel de detaliu, până când activităţile rezultate pot fi executate de un singur agent. Un exemplu de descompunere funcţională este prezentată în figura 3.5.
Alocarea activităţilor este o problemă de optimizare a atribuirii acestora unor agenţi umani sau maşini din sistem, pornind de la constrângeri legate de context.

Alocarea activităţilor conduce la două alte probleme:

- stabilirea ***fluxurilor de materiale*** între agenţi, care să asigure desfăşurarea proceselor fizice de prelucrare;
- stabilirea ***fluxurilor de informaţii*** între agenţi, necesare funcţiilor de monitorizare, control şi supervizare.

Rezolvarea problemei conduce la seturi de strategii de alocare şi la soluţii suboptimale, ţinând cont că problema împărţirii în funcţii şi subfuncţii este şi ea o problemă neoptimizată.

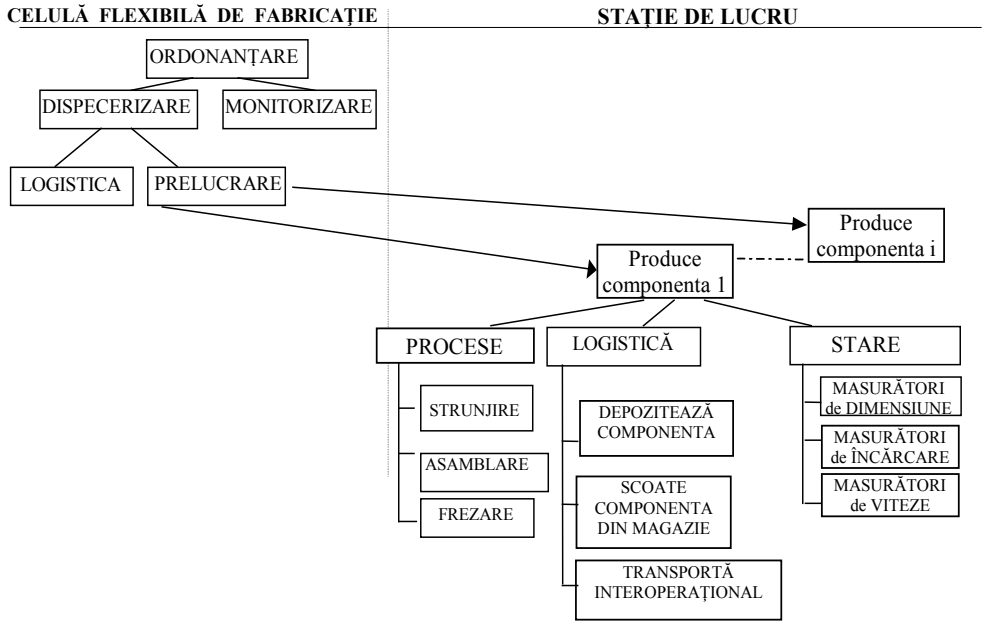


Fig.3.5. Exemplu de descompunere funcţională

Trebuie ţinut cont că în sistemele complexe om-maşină optimizarea nu este în sine o cerinţă de bază, mult mai robustă fiind soluţia care propune ca alocarea activităţilor să fie un proces dinamic, depinzând de natura activităţilor dar şi de starea curentă a agenţilor din sistem.

Matematicile discrete şi aplicaţiile lor în modelarea proceselor cu evenimente discrete se constituie într-un cadru conceptual şi analitic pentru reprezentare, reţelele Petri Colorate oferind posibilităţi explicite în acest sens [Jensen,93].

În conformitate cu rolurile, nucleul cel mai de jos al descompunerii funcţionale pentru o aplicaţie particulară în care este implicat şi agentul uman, se poate utiliza un model de execuţie a activităţii de către o singură resursă.

Modelul propus utilizează cinci niveluri pentru a defini rolurile (figura 3.6.):

- evaluare situaţii;
- fuziune informaţii procesare activităţi;
- interpretare comandă ;
- selectare răspuns.

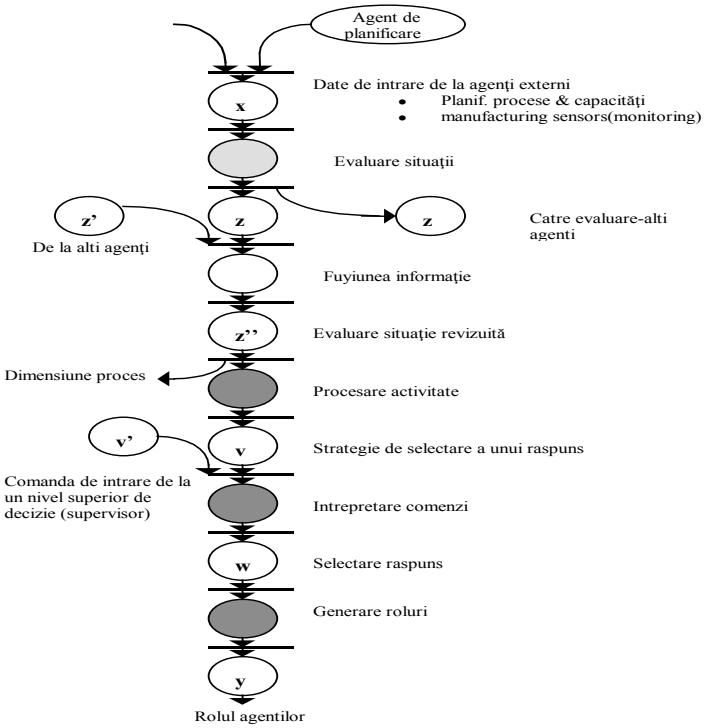


Fig. 3.6. Modelul rolurilor structurat pe cinci niveluri

3.1.4. DEFINIREA NOŢIUNII DE FABRICATOR

Într-un sistem de fabricaţie denumit în continuare ***fabricator*****,** fabrica virtuală asigură funcţiunile de management de procese şi resurse pentru fabrica fizică (**PF**-Physical Factory) prin intermediul sistemului automat de control (AS-Automation System). Din analiza funcţiunilor din fabricator acoperite de CAM şi PP&C (cap.1) se pot separa funcţiunile implementate soft de acelea implementate prin utilizarea de elemente non-software, echipamente (roboţi, maşini unelte CNC, AGVs, etc.) şi se poate face referire la arhitecturi software şi la modul în care acestea se integrează cu entităţile fizice pe care le controlează şi conduc.

Pentru a realiza serviciile alocate, fabrica virtuală utilizează modele care descriu entităţile din fabricator şi funcţionalitatea lor. Modelele sunt utilizate pentru evaluarea informaţiei senzoriale (monitorizare, diagnoză) şi crearea planurilor de acţiune (planificare procese, ordonanţare şi re-ordonanţare) care sunt transmise sistemului automat de control pentru execuţie.

Fabrica virtuală are la bază un sistem de management şi modelare. Sistemul de management evaluează informaţiile primite de la sistemul de modelare precum şi

din procesele fizice de fabricaţie sau asociate şi decide asupra acţiunilor ce trebuie îndeplinite pentru a satisface sistemul global de scopuri.

Scopurile fabricatorului sunt influenţate de mediul de afaceri în care acesta operează. Sistemul de modele include modele ale elementelor din **PF** (facilităţi, maşini), modele de procese de afaceri şi fabricaţie, modele de control şi modele de suport decizie.

Din definiţia fabricii virtuale şi din "galaxia CIM" (cap.1) se observă că elementul esenţial pentru universul activităţilor inginereşti dintr-un sistem de fabricaţie la care VF nu face referire este universul CAD (proiectare produse şi procese); rezultatele acesteia sunt însă preluate de VF din baza de date cu produse şi procese rezultate în urma proiectării în mediul de Inginerie Concurentă.

Sistemul de automatizare bazat pe calculator, **CBAS** (Computer Based Automation System) sau **AS** are la bază o platformă de integrare care "asociază" proceselor virtuale de fabricaţie, ordonate prin planul de ordonanţare, activităţi cadru (patternuri de activităţi) spaţio-temporale. În [Raulefs,94] se propune o organizare a AS, pe componentele principale: ***Servicii Distribuite***, ***Staţii de lucru*** (logice) şi ***Modele şi servere de aplicaţii***.

Sistemul de fabricaţie efectiv, fabrica fizică **MCS** (Manufacturing Controlled System) sau **PF** (Physical Factory) reprezintă totalitatea facilităţilor eterogene de fabricaţie:

^{P} **R** - setul de maşini implicate în Prelucrare (maşini de strunjire, frezare, maşini de măsurat în coordonate etc.) ;
^{D} **R** - setul de Dispozitive (elemente de prindere, de fixare, de apucare);
^{M} **R** - setul de dispozitive de Manipulare şi transfer (roboţi, manipulatoare etc.);
^{T} **R** - setul de sisteme de Transport (ex. AGV, AVM, transportoare);
^{S} **R** - setul de elemente de depozitare-Stocare (ex.magazii automate, depozite Interoperaţionale etc.);
^{U} **R** - setul de agenţi Umani (de exemplu: experţi, operatori, muncitori);
^{C} **R** - setul de produse aflate în curs de realizare "*work-in-progress*" (materii prime, subansamble etc.)

Definiţia 1*: FABRICATOR-ul este un agent autonom care oferă produse şi servicii pieţei de afaceri. El constă din trei subsisteme ierarhice: subsistemul fabrica fizică, constând în totalitatea echipamentelor programabile din sistemul de fabricaţie, subsistemul fabrica virtuală şi subsistemul claselor de sisteme de automatizare prin care fabrica virtuală interacţionează cu fabrica fizică.*

Definiţia 2*: FABRICATOR-ul poate fi formal definit ca un meta-sistem deschis aflat într-un univers de afaceri, constând dintr-o mulţime de resurse organizate în fabrica fizică, un model de fabrică virtuală, mulţimea claselor de sisteme de*

automatizare prin care fabrica virtuală interacţionează cu fabrica fizică, şi o structură scop :

F = (R, M_{FV}, S, G) unde:

R = *mulţimea de resurse din fabrica fizică*
M_{FV} = *modelul fabricii virtuale*
S = *mulţimea claselor de sisteme de automatizare*
G = structura scop

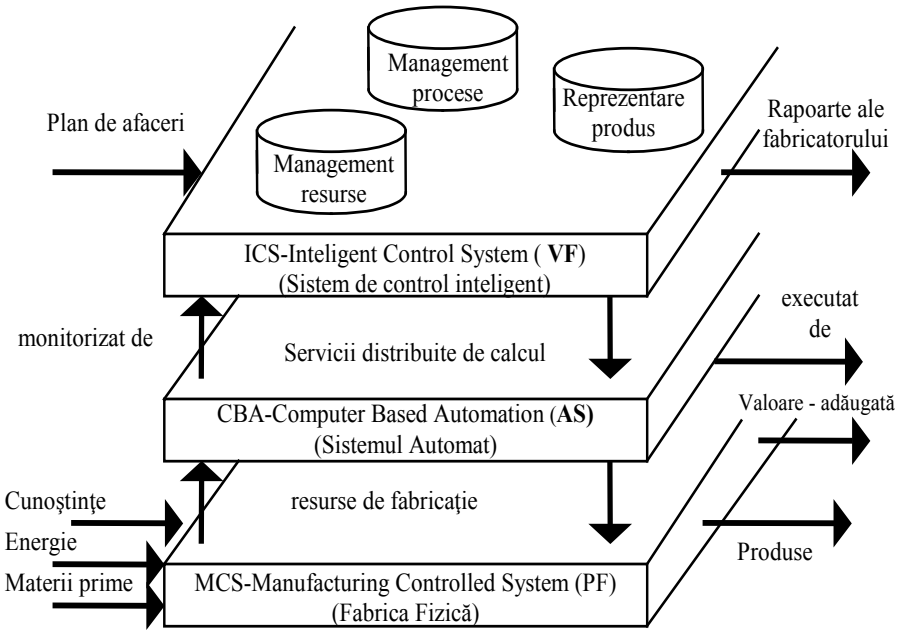


Fig. 3.7. Conceptul de Fabricator

3.2. PLATFORMA MULTI - DIMENSIONALĂ PENTRU MODELARE

Pentru a captura caracteristicile statice şi dinamice ale sistemelor industriale este necesară utilizarea diferitelor clase de modele.

Procesul de modelare poate fi divizat în două faze majore:

- descriere de nivel-înalt;
- descriere detaliată de sistem.

Modelarea sistemelor de fabricaţie diferă de modelarea sistemelor informatice prin accea că necesită o reprezentare multidimensională, produs, proces, sistem, fiecare din aceste reprezentări presupunând, la rândul ei o reprezentare multidimensională.

Fiecare model din ***fabricator*** trebuie să asigure legătura dintre nivelul fabricii virtuale şi facilităţile aflate la nivelul fabricii fizice, prin care să se asigure posibilitatea simulării diverselor configuraţii de fabricaţie şi politicile de alocare a resurselor.

Cadrul de multimodelare trebuie să raspundă următoarelor cerinţe de proiectare colaborativă:

- ***să se constituie într-un sistem deschis***, care să permită integrarea diferitelor modele de domeniu specific;
- ***să permită procesarea concurrent – distribuită***, atât între diferite niveluri ierarhice (atelier, celulă, staţie de lucru, echipament) cât şi la acelaşi nivel ierarhic;
- ***sa permită rafinarea modelelor corespunzător diferitelor niveluri de complexitate.***

La nivel de formulare de cerinţe scopul modelării este acela de a defini specificaţiile pentru un sistem de fabricaţie în termeni de funcţionalitate a sistemului, structură a sa, comportare, în vederea proiectării şi implementării sistemului de control.

Se propune [Stanescu,97] un cadru de modelare 4D (patru dimensional) orientat: ***Funcţional, Structural, Comportamental şi Contextual*** pentru analiza, proiectarea şi implemenatrea unui sistem de control al unui sistem de fabricaţie, ca nucleu de cercetare privind sinergia diverselor clase de modele şi a cerinţelor de interoperabilitate ale acestora.

Primele două dimensiuni se pot asocia unei descrieri de nivel înalt, iar celelalte două unei descrieri detaliate a sistemului. În concordanţă cu metodologia top-down proiectarea contextuală a sistemului automat are ca prim nivel Managementul Sistemului de Control (MCS), urmată de Sistemul Automat de Control (ACS), ajungând până la modelul extins al maşinii, utilizat pentru simularea în mediu virtual de aplicaţii, pentru optimizare rute, fluxuri de materiale şi subansamble.

Din considerente de calitate a proiectării sistemului automat se propune o metodologie de proiectare colaborativă în spirală, astfel încât analiza, proiectarea şi reproiectarea să se poată face prin iteraţii succesive, la diverse niveluri de rafinare a modelului, aşa cum se prezintă în figura 3.9.

Semnificaţia proiectării în spirală poate fi asociată şi cu un proces de micşorare a incertitudinii pe măsura parcurgerii spiralei între cele patru dimensiuni şi prin coborâre în ierarhie.

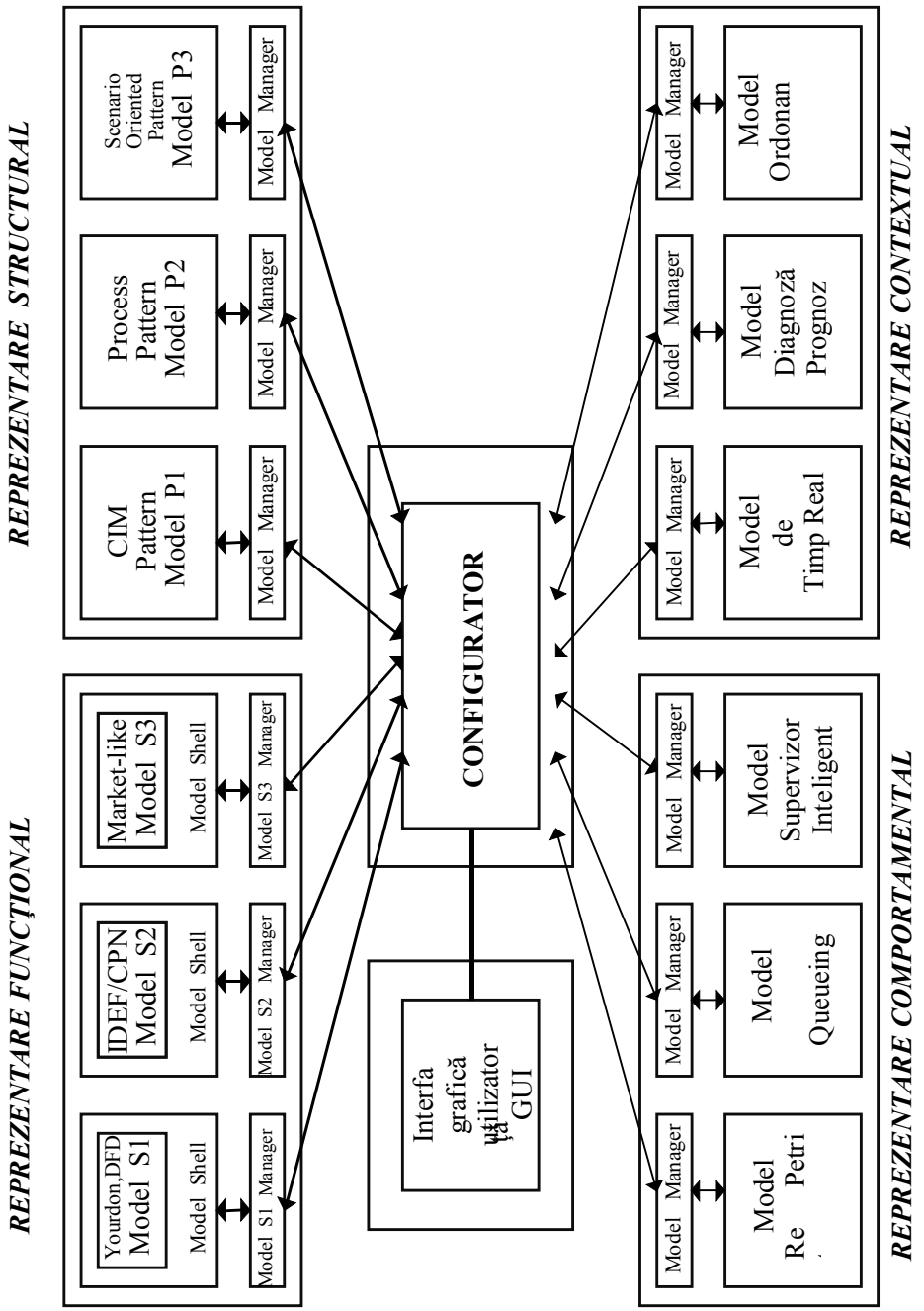


Fig. 3.8. Cadru 4D de multimodelare

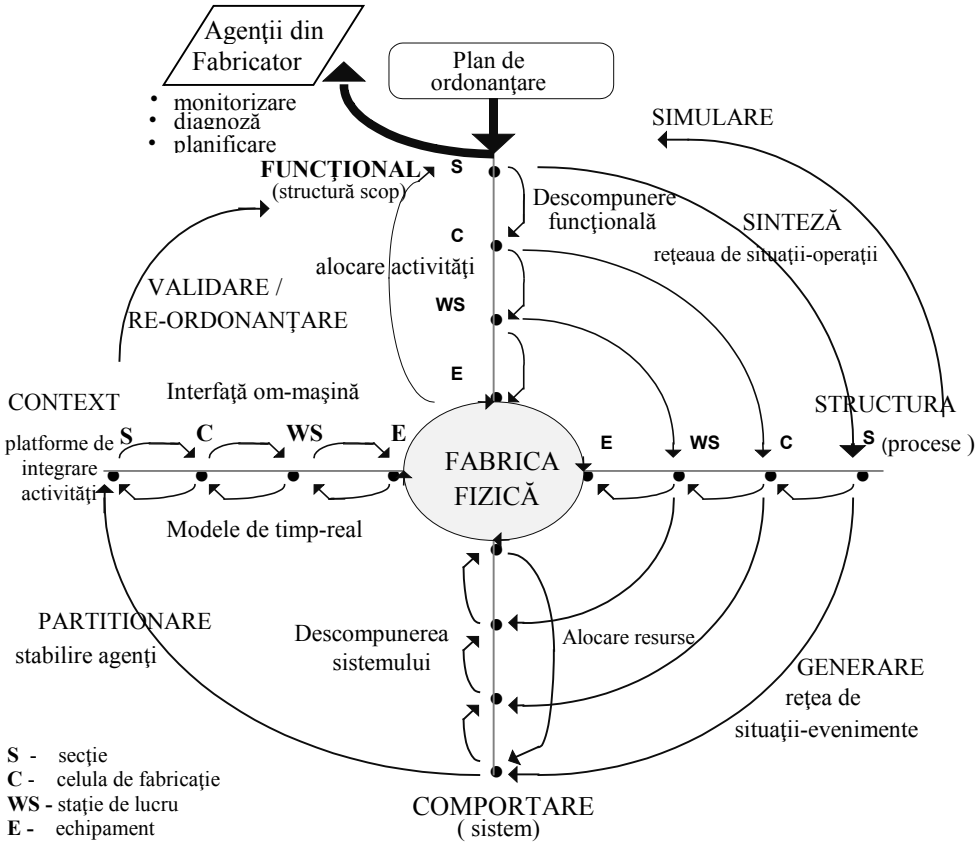


Fig. 3.9. Metodologie integrată de proiectare

3.2.1. DIMENSIUNEA – REPREZENTARE FUNCŢIONALĂ

Modelele funcţionale arată ***ce trebuie să facă sistemul***, fară a arăta cum şi când. În esenţă, modelul funcţional constă într-o reţea de unităţi de procesare denumite procese sau funcţii. Fiecare proces transformă datele de intrare în rezultate de ieşire, care pot fi, la rândul lor, intrări pentru alte procese. Problema iniţială în rezolvarea problemei proiectării AS este asociată cu definirea/precizarea

structurii scop. Agenţii – oameni sau maşini inteligente – contribuie fiecare la executarea diferitelor funcţii, secvenţial sau concurent.

Descompunerea funcţională trebuie să sprijine alocarea de funcţiuni fiecărui agent, astfel încât încărcarea rezultată să nu depăşească capacitatea fiecăruia dintre ei. Descompunerea funcţională identifică ***domenii*** ale sistemului de fabricaţie (SF).

Un domeniu este un subset al unui SF definit prin: ***obiective specifice*** de domeniu, satisfăcute de ***procese specifice*** de domeniu. Procesele specifice de domeniu sunt compuse din ***procese de afaceri***, care reprezintă structuri intermediare în ierarhia descompunerii funcţionale.

La nivelul cel mai de jos al ierarhiei apar ***activităţile*** din SF, care sunt identificate ca taskuri care "consumă" funcţia de intrare şi produc funcţia de ieşire, necesitând alocare de resurse şi timp pentru execuţie [Edelweiss,95]. Descompunerea este concludentă atunci când toate activităţile din SF sunt clar definite. Gradul de detaliere al modelului determină gradul în care proiectarea sistemului poate deriva direct din modelare.

Pentru exemplificare se va prezenta metodologia Yourdon. Urmărind această metodologie se creează un ***model esenţial*** care are la bază alte două modele create pe baza diagramelor de context şi a metodologiei DFD (Data Flow Diagram).

Metodologiile folosesc un formalism grafic asemănător pentru crearea modelului funcţional, fiind de fapt ***grafuri orientate***, cu trei tipuri de noduri: ***procese, agenţi şi unităţi de păstrare de date***. Arcele orientate sunt denumite ***dataflow*** şi conectează perechi de noduri, fiecare arc reprezentând un flux de informaţii dintre un nod sursă şi un nod destinaţie. Informaţiile sunt înglobate în unităţi discrete denumite ***jetoane***, iar jetoanele care fac parte dintr-un flux de date stabilit sunt omogene. Conţinutul jetoanelor poate fi definit formal printr-o structură textuală, denumită dicţionar de date.

Din punctul de vedere al analizei funcţionale, ***procesele*** sunt paşi atomici de calcul care necesită timp zero pentru execuţie şi nu sunt funcţie de stare, depinzând numai de datele de intrare. Acţiunile pot fi specificate formal într-un dicţionar de date asociat. ***Agenţii*** modelează entităţile din sistem şi au rolul de a transmite şi/sau recepţiona jetoane către/de la procese. ***Unităţile de păstrare de date*** au rolul de a păstra jetoanele pentru utilizări ulterioare şi sunt conectate numai la procese. Din punctul de vedere al relaţiilor dintre procese, acestea pot fi ***puternic cuplate*** dacă există legătură de flux de date direct între ele sau ***slab cuplate*** dacă legătura este facută printr-o unitate de păstrare de date.

Modelul de sistem, specificat prin diagrama de context, figura 3.10., poate fi considerat un model detaliat al modelului de referinţă (figura 1.12.).

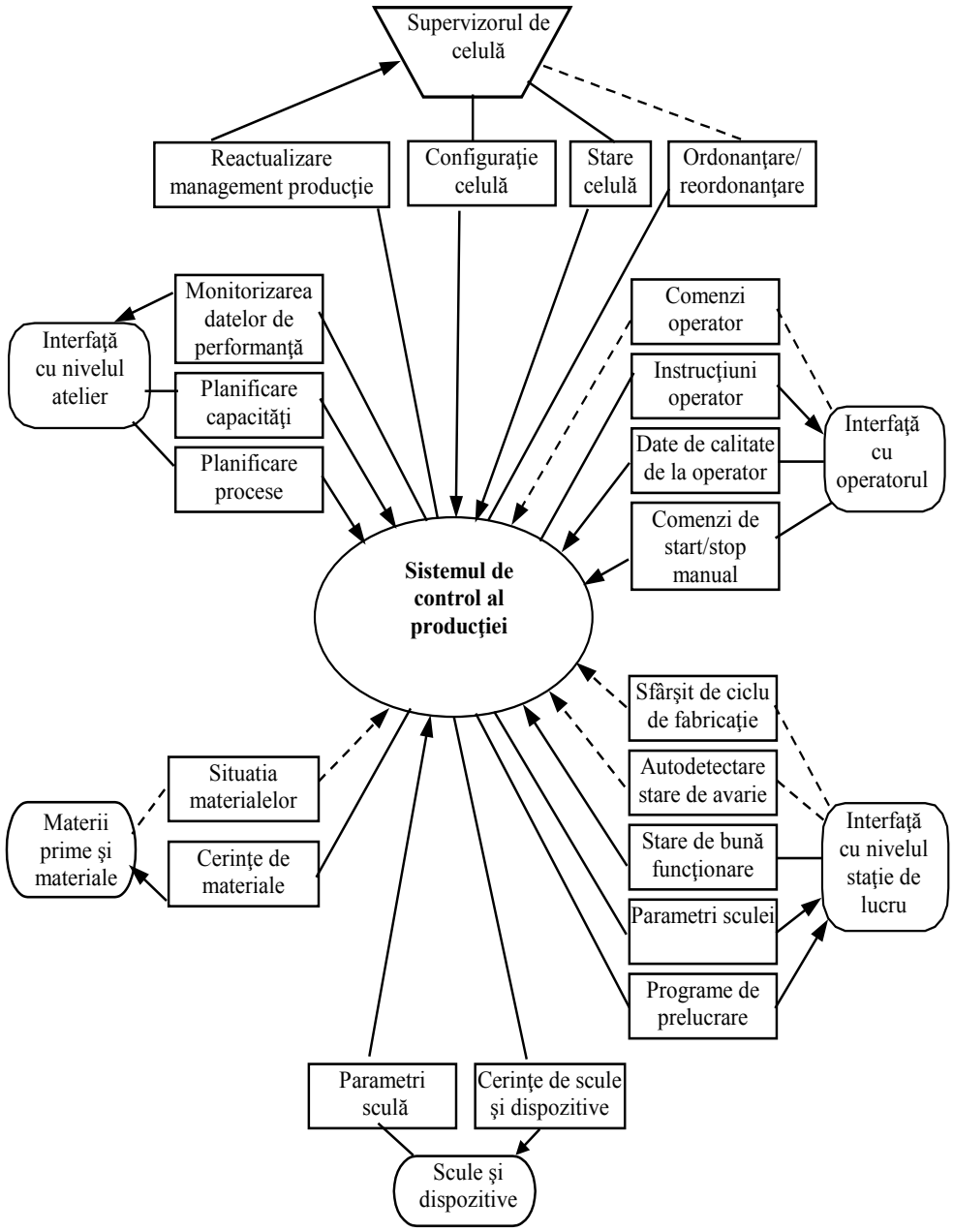


Fig. 3.10. Diagrama de context pentru sistemul de control al producţiei

Diagrama conţine supervizorul de celulă care urmăreşte operaţiunile sistemului de control al producţiei şi poate interveni activ, atunci când este necesar, prin comanda unei noi ordonanţări/re-ordonanţări, prin modificarea criteriilor de ordonantări sau prin revizuirea rapoartelor de stare recepţionate. Informaţiile referitoare la solicitarea de materiale şi scule/dispozitive precum şi starea lor sunt necesare pentru controlul producţiei şi realizarea ordonanţării.

Diagrama de context este însoţită de o listă care oferă o descriere a tuturor evenimentelor din sistem care sunt luate în considerare, iar sistemul de control trebuie să ofere răspuns.

Modelul DFD din figura 3.11. descrie comportarea sistemului ca reacţie la evenimentele din mediul în care acesta evoluează.

Modelul din figura 3.11. trebuie extins pentru a obţine un nivel superior de detaliere. Modelul creat permite analiza de sistem şi este utilizat în faza de implemenatre, când fiecărui element al modelului îi este asociată o resursă hardware sau software. Pentru a susţine interconectarea în timp real a sistemului de control cu multitudinea de echipamente, care utilizează diverse standarde, s-au dezvoltat interfeţe maşină specifice.

Pentru a descrie funcţionalităţile sistemului de fabricaţie în termeni de activităţi de producţie şi relaţii între acestea, se poate utiliza un model creat pe baza metodologiei IDEF0 – Integration Definition for Function Modeling [Standard,93].

IDEF0 tratează complexitatea activităţilor de fabricaţie prin adoptarea unei ierarhii de niveluri, astfel încât pe măsura coborârii în ierarhie, gradul de detaliu creşte progresiv. La nivelul de vârf ierarhia IDEF0 conţine o singură activitate.

Descrierea procesului de fabricaţie utilizând specificaţiile limbajului grafic SADT (*Structural Analyses and Design Techniques*) denumit în USA, IDEF şi "completarea" sa cu informaţii precise legate de comportarea sistemului permit translatarea automată într-o reţea CPN (Colored Petri Net), reţea ce poate fi simulată şi poate conduce la un model executabil de sistem [Jensen,95], asigurându-se interoperabilitatea între formalismul grafic IDEF0 şi formalismul reţelelor Petri colorate.

Al treilea tip de model funcţional, legat de piaţă, este deosebit de important pentru îndeplinirea cerinţelor de sistem adaptiv. Dacă proiectarea în cadrul echipei de inginerie concurentă conduce la un plan flexibil de procese, modelarea orientată piaţă [Lindberg,93] descrie cadrul integrat necesar unui control adaptiv referitor la cerinţele pieţii, externe sistemului de fabricaţie.

Dacă agenţii de prelucrare din sistem iau decizii în funcţie de obiectivele de fabricaţie pe care trebuie să le îndeplinească, agenţii de tip resursă iau decizii bazate pe sistemul de evaluare a pieţii. Obiectivul agenţilor de prelucrare este acela

de a-şi atinge sructura scop cu costuri minime, iar a celor de tip resursă este acela de a minimiza costurile necesare satisfacerii cerinţelor agenţilor de prelucrare.

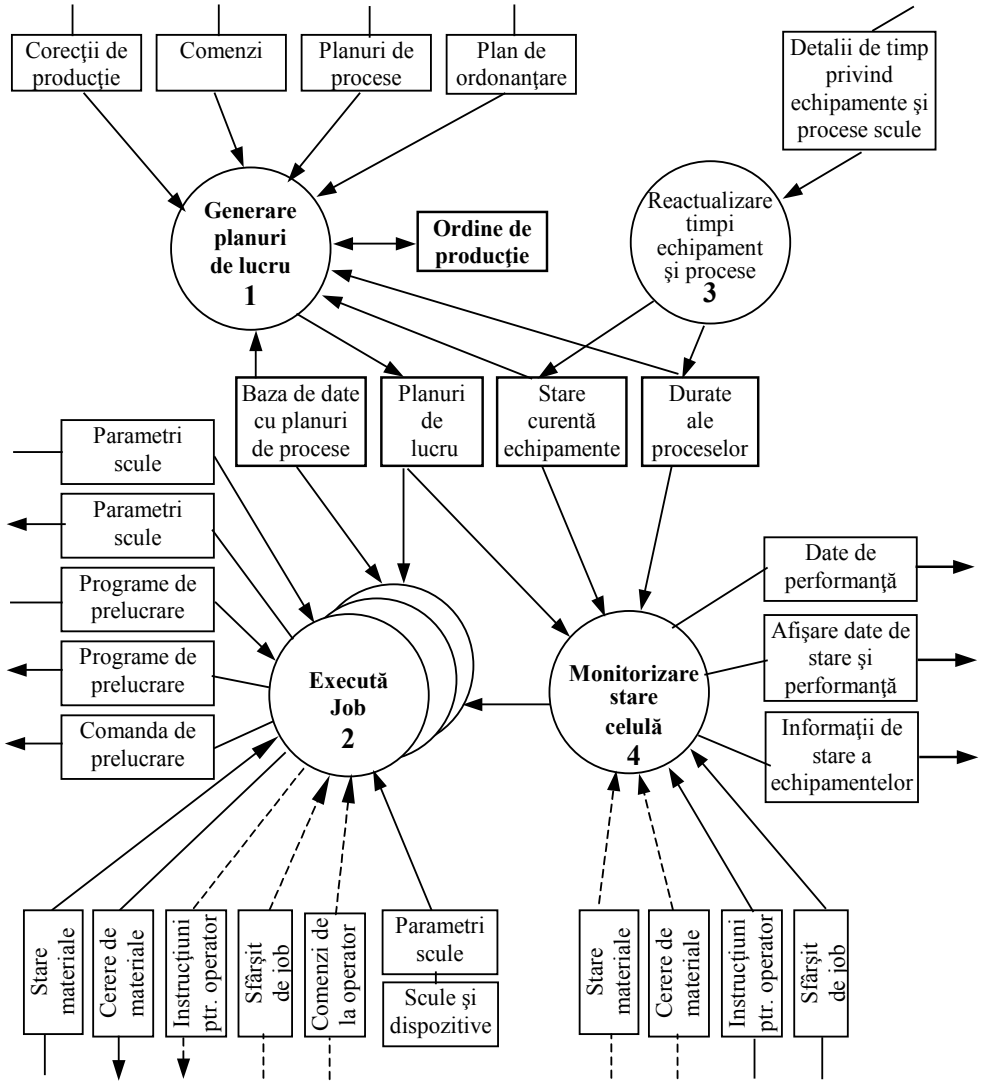


Fig. 3.11. Model comportamental (nivel 1.1)

3.2.2. DIMENSIUNEA – REPREZENTARE STRUCTURALĂ

Se poate observa că, ceea ce caracterizează proiectarea mai mult decât elementele în sine, sunt ,,tiparele-cadru/pattern-uri" care le leagă. Într-un mediu orientat pe obiecte pentru proiectarea integrată a aplicaţiilor de fabricaţie, elementele sunt obiecte, iar "pattern-urile" indică grupuri de obiecte care sunt legate prin anumite relaţii fixe şi cooperează. Problema la care se face referire este aceea a proiectării arhitecturii de control a sistemului automat bazat pe niveluri de module concurente. Acestea trebuie să poată fi implementate într-o arhitectură de calcul distribuit care să ofere cadrul pentru un proces evoluat de dezvoltare care să transforme un prototip de proiectare logică într-o proiectare fizică.

Utilizarea pattern-urilor permite structurarea aplicaţiilor prin divizarea lor în activităţi independente, ce pot fi efectuate secvenţial sau paralel, predeterminat.

Pattern-urile sunt structuri de proiectare predefinite care joacă rolul de ,,blocuri constructive" având rolul de a sprijinind dezvoltarea rapidă de aplicaţii software.

Proprietăţile fundamentale ale pattern-ului sunt [Gamma,93]:

- oferă o schemă predefinită pentru implementarea unei structuri particulare sau a unor principii funcţionale pentru sistemele software, prin descrierea diferitelor sale părţi precum şi a colaborărilor şi a responsabilităţilor;
- capturează experienţa de proiectare existentă, validată;
- identificatori, nume şi abstractizare peste nivelul claselor şi instanţelor;
- asigură înţelegerea şi un vocabular comun pentru principiile de proiecatre;
- sunt blocuri constructive reutilizabile pentru dezvoltarea de software;
- pot fi structuri independente de domeniu [Gamma,94], sau pot adresa aspecte specifice de domeniu "G++"[Menga,93];
- adresează atât aspecte funcţionale cât şi non-funcţionale ale proiectării software.

Pattern-urile acoperă numeroase aspecte relevante ale arhitecturii software. Ele au în vedere structura unei aplicaţii, componentele ei, relaţiile dintre ele şi proprietăţile funcţionale şi non-funcţionale. Literatura de specialitate recunoaşte că aceste aspecte au o mare importanţă pentru succesul realizării de software [Reenskaug,92; Buschmann,93; Garlan,93].

Descrierea pattern-urilor

Conform principiilor definite in [Alexander,79] se pot distinge trei părţi esenţiale ale unui pattern:

- ***context:*** o situaţie particulară;
- ***problemă:*** un sistem de forţe care apar în context;
- ***soluţie:*** o configuraţie spaţială care permite proiectantului să rezolve o problemă.

Toate pattren-urile dintr-un sistem sunt descrise într-un mod unitar, astfel:

- ***nume***: esenţa pattern-ului;
- ***raţiunea***: motivaţia dezvoltării paternului;
- ***aplicabilitatea***: este prezentată atunci când se utilizează pattern-ul;
- ***clasificarea***: clasificarea se face în concordanţă cu regulile ce vor fi prezentatate în paragraful următor;
- ***descrierea***: sunt descrişi participanţii şi colaboratorii în pattern, ca şi responsabilităţile lor şi relaţiile fiecăruia cu ceilalţi;
- ***diagrama***: oferă o reprezentare grafică a structurii pattern-ului;
- ***comportarea dinamică***: în funcţie de specific, se prezintă comportarea dinamică;
- ***metodologie***: listează metodologia, respectiv paşii necesari pentru construirea unui pattern;
- ***implementare***: ghid de implementare a pattern-ului;
- ***variante***: descrierea variantelor posibile ale pattern-ului;
- ***exemple***: exemple de utilizare a pattern-ului;
- ***discuţii***: prezentarea constrângerilor de aplicare a pattern-ului;
- ***altele***: referinţe către pattern-uri aflate în relaţie.

Pentru a ghida selectarea pattern-ului pentru o situaţie dată, sistemul pattern-uri include o schemă după care sunt clasificate toate acestea.

Schema de clasificare constă în trei categorii de criterii sau rezultate de proiectare care joacă un rol important în dezvoltarea de software:

- **Granularitatea:** dezvoltarea unui sistem software presupune utilizarea diferitelor niveluri de abstractizare, pornind cu structura de bază a aplicaţiei şi sfârşind cu realizarea prin proiectare a unei structuri particulare. Sunt precizate trei niveluri de granularitate:
  - **cadrul arhitectural**: reprezintă paradigmele fundamentale pentru structurarea sistemului software;
  - **pattern-uri de proiectare**: descriu scheme pentru structurarea subsistemelor şi componentelor arhitecturii software, precum şi relaţiile lor;
  - **expresii**: descriu cum se implementează componentele particulare ale unui subsistem şi funcţionalitatea componenetelor, sau relaţiile cu alte componenete în cadrul unui proiect dat.
- **Funcţionalitatea:** pattern-ul serveşte drept cadru pentru implementarea unei funcţionalităţi particulare. Pot fi distinse următoarele categorii funcţionale:
  - ***crearea obiectelor:*** pattern-urile pot specifica cum se creează instanţe particulare ale structurilor de obiecte complex recursive sau agregate;

- ***orientarea comunicaţiilor dintre obiecte:*** pattern-urile pot descrie cum se organizează comunicaţiile în cadrul unui set de obiecte care colaborează care poate fi dezvoltat independent sau de la distanţă;
- ***accesul la obiecte:*** patternu-rile pot descrie cum se realizează accesul într-un mod sigur/protejat la serviciile şi la starea obiectelor partajate sau remote, fără a viola starea sau comportarea încapsulată;
- ***organizarea calculului de funcţii complexe:*** pattern-urile pot specifica cum se distribuie responsabilităţile între obiectele care cooperează pentru a rezolva funcţii şi sarcini complexe.

❑ **Principii structurale:** pentru a-şi realiza funcţionalitatea, pattern-urile se bazează pe anumite principii structurale:

- ***abstractizare:*** pattern-ul oferă o imagine abstractă sau generalizată a particularului;
- ***încapsularea:*** pattern-ul încapsulează detalii ale unui obiect particular, componentă sau serviciu, pentru a anula dependenţele sale de clienţi sau pentru a proteja aceste detalii accesului;
- ***separarea sarcinilor:*** pattern-ul distribuie responsabilităţi specifice unor obiecte separate cu scopul de a rezolva sarcini specifice sau pentru a furniza anumite servicii;
- ***cuplare şi coeziune:*** pattern-ul permite anularea sau relaxarea numai a relaţiilor şi dependenţelor structurale şi de comunicaţii între obiecte puternic cuplate.

Domeniul CIM, prin natura sa, include un număr mare de elemente eterogene, şi se constituie într-un mediu propice definirii unor patternuri de arhitectură şi unui cadru de clase, într-o aplicaţie de domeniu specific [Menga,94], conform figurii 3.12.

3.2.3. DIMENSIUNEA – REPREZENTARE COMPORTAMENTALĂ

Sistemele de fabricaţie flexibilă sunt, din punct de vedere structural, sisteme din clasa celor cu evenimente discrete, mai ales din punctul de vedere al ordonanţării, care constituie o parte esenţială a sarcinii de conducere, alături de monitorizare şi diagnoză. Caracteristicile de bază ale acestei clase de sisteme sunt: evoluţia bazată pe evenimente, asincronismul, relaţiile secvenţiale între evenimente, concurenţa, existenţa conflictelor de alocare a resurselor (structurale şi/sau efective), excluderea mutuală, nedeterminismul, posibilitatea existenţei blocajelor – toate acestea fiind evidenţiabile prin intermediul formalismului SED.

Pentru controlul avansat al sistemelor de fabricaţie integrate în structuri de întreprindere virtuală este necesară abordarea sistemului ca un sistem cu evenimente discrete.

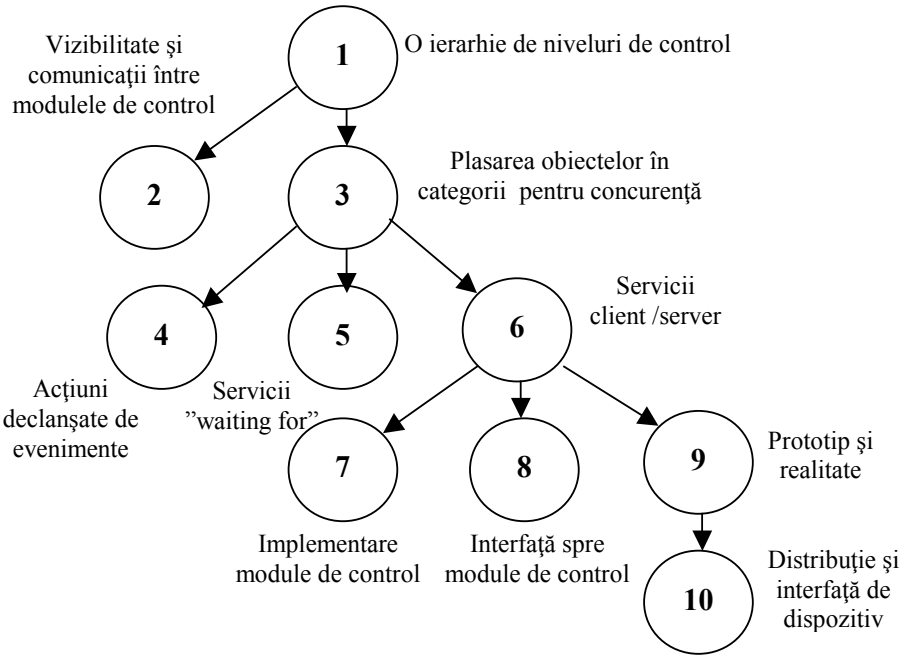


Fig. 3.12. Pattern-uri de arhitectură

A. Prezentare conceptuală a Sistemelor cu Evenimente Discrete

Definiţie: *Un sistem cu evenimente discrete (SED) este un sistem dinamic cu un spaţiu al stărilor discret şi cu traiectorii de stare continue pe porţiuni; momentele la care intervin tranziţiile de stare, ca şi tipul tranziţiilor, sunt de regulă impredictibile [Ramadge,89].*

Tranziţiile de stare ale unui SED sunt numite *evenimente şi* pot fi etichetate cu elementele unui alfabet oarecare $\sum$. Aceste *etichete* desemnează uzual fenomenele fizice care produc modificările de stare.

Modelarea unui proces ca SED poate să aibă două scopuri:

- studiul comportării sistemului;
- conducerea sistemului astfel încât acesta să aibă o anumită evoluţie.

In acest din urmă caz, conducerea se face prin generarea, la intrarea sistemului, a unei anumite secvenţe ordonate de evenimente. Există mai multe tipuri de formalisme cu ajutorul cărora se poate face modelarea proceselor SED, în ***funcţie de obiectivul pentru care se face modelarea***.

Clasificare a modelelor SED [Giua,92]:

- ***modele logice*** – se bazează pe ipoteza simplificatoare a ignorării momentului producerii unui eveniment şi consideră importantă numai ordinea în care evenimentele se produc. Această ipoteză este justificată atunci când modelele sunt utilizate în scopul studierii proprietăţilor dinamice independente de aspecte temporale.
- ***modele temporizate sau modele de performanţă*** – care sunt potrivite pentru studiul proprietăţilor care depind explicit de timp. Aceste modele pot fi clasificate în continuare în modele:
  - *ne-stochastice* – momentele şi intervalele de timp sunt cunoscute a-priori;
  - *stochastice* – momentele şi intervalele de timp nu sunt cunoscute a-priori, din cauza existenţei unor întârzieri sau apariţiei unor evenimente aleatoare.

Următoarele aspecte au fost propuse a fi utilizate pentru evaluarea şi compararea diferitelor tipuri de modele SED [Inan,88]:

Puterea descriptivă;

Posibilitatea specificării mulţimii traiectoriilor de stare admisibile. Se poate utiliza:

- *o structură de tranziţie de stare* (automate de stare finite, reţele Petri, automate Büchi);
- *un set de ecuaţii* (modele booleene, procese secvenţiale care comunică, procese recursive finite);
- *calcule logice* (logica temporală).

Diverse modele pot modela mai bine sau mai puţin bine acelaşi comportament, iar acest aspect calitativ este denumit *puterea de reprezentare* a modelului. În general fiecare formalism generează clase de limbaje specifice, adică fiecare model are puterea proprie limbajului pe care-l generează. Clasele de operatori proprii fiecărui tip de modele definesc *puterea algebrică* a acestora.

Evaluarea performanţei

Odată ce este sigur că modelul a capturat trasăturile esenţiale ale sistemului, acesta poate fi utilizat la verificarea proprietăţilor de sistem dorite şi a comportării sistemului.

Acest gen de verificări se fac astfel:

a. se construieşte modelul prin „translatarea” proprietăţilor de sistem dorite în proprietăţi ale modelului;
b. se utilizează metode algoritmice, analitice sau de simulare pentru a verifica dacă modelul are sau nu proprietăţile dorite;

c. problema dificilă este cea a complexităţii calculelor; uzual cu cât este mai mare puterea de modelare a unui formalism, cu atât modelele obţinute sunt mai dificil de analizat.

Implementarea politicii de conducere

Modelarea şi analiza proceselor reprezintă paşi preliminari în studiul DES, scopul fiind de regulă controlul unui DES prin modificarea traiectoriilor de stare, astfel încât sistemul să aibă comportarea dorită.

B. Caracteristicile modelului SED pentru sisteme de fabricaţie

Din punctul de vedere al conducerii şi planificării ***fabricatorul*** poate fi considerat ca un set de *resurse (fabrica fizică)* care asigură realizarea proceselor de fabricaţie în ordinea stabilită prin planul de ordonanţare, respectând un set de *specificaţii (stabilit la nivelul AS)*.

Obiectivul unui fabricator este de a obţine producţia planificată, într-un interval de timp prestabilit, în condiţiile maximizării funcţiei obiectiv (cap.4). Planificarea şi ordonanţarea ţin cont de: operaţiile tehnologice ce pot fi realizate de o anumită resursă, de duratele operaţiilor, de ordinea tehnologică, de disponibilitatea resursei, etc. exprimate în termeni de evenimente: start/stop operaţie, resursa_OK, resursa_defectă, resursa_în_reparaţie, sosire_ piesa, etc., şi condiţii.

Caracteristicile unui sistem de fabricaţie ca sistem cu evenimente discrete sunt:

- evoluţia bazată pe evenimente;
- asincronismul;
- secvenţialitatea;
- concurenţa şi prezenţa conflictelor;
- excluderea mutuală;
- nedeterminism .

C. Modelarea SED cu reţele Petri

Printre cele mai răspândite instrumente de modelare ale SED se numără automatele de stare, reţelele Petri şi Grafcet-ul. În cele ce urmează vor fi prezentate câteva considerente care explică alegerea formalismului reţelelor Petri ca instrument de modelare pentru sinteza politicii de conducere în timp real a sistemelor de producţie. Tehnicile de analiză ale reţelelor Petri pot fi împărţite în următoarele categorii [Giua,92]:

- ***analiza prin enumerare*** – necesită construirea unui arbore de acoperire (sau graf de marcaje) care include toate marcajele accesibile şi secvenţele de tranziţii prin intermediul cărora sunt atinse ;

- ***analiza prin transformare*** – o reţea poate fi transformată, aplicând anumite reguli, într-o nouă reţea care păstrează proprietăţile dorite, având ca scop obţinerea unei reţele mai uşor de analizat. Metodele de reducere [Zhou,93] permit simplificări structurale ale reţelei cu asigurarea păstrării proprietăţilor iniţiale ale acesteia.
- ***analiza structurală*** – permite demonstrarea anumitor proprietăţi aproape independente de marcajul iniţial. Analiza structurală se poate baza pe strudiul ecuaţiilor de stare sau pe studiul grafului reţelei.
- ***analiza prin simulare*** – este utilizată pentru reţelele care interacţionează cu mediul extern.

Pentru modelele logice se pot defini diferite clase de limbaje generate de RP respective (limbajele RP). Deşi limbajele RP nu au toate proprietăţile de închidere ale limbajelor regulate, ele reprezintă totuşi supersetуri ale acestora şi, prin introducerea unui număr de operatori, permit definirea unei algebre [Caramihai,97].

Reţele Petri ordinare

Reţelele Petri sunt grafuri orientate bipartite. Au două tipuri de noduri: poziţii (P) – reprezentate prin cercuri şi tranziţii (T) – reprezentate prin segmente de dreaptă, unite prin intermediul arcelor. Nodurile de fiecare tip sunt în număr finit şi nenul (există RP degenerate care pot să nu aibă nici o poziţie sau nici o tranziţie, dar nu reprezintă cazuri interesante din punctul de vedere al acestei lucrări). Arcele trebuie să aibă câte un nod la fiecare capăt, şi fiecare arc trebuie să unească două noduri de tip diferit.

Definiţie [Zhou,93] ***: O reţea Petri*** **(RP)** marcată este o structură

$$\mathbf{Z = (P_A \cup P_B \cup P_C, T, I, O, m_0),}$$ unde:

$\mathbf{P_A}$ = mulţimea poziţiilor reprezentând operaţii

$\mathbf{P_B}$ = mulţimea poziţiilor reprezentând resurse fixe (al căror număr este fixat prin specificaţii de sistem) roboţi, maşini, elemente transportoare

$\mathbf{P_C}$ =mulţimea poziţiilor reprezentând resurse variabile, palete, elemente de prindere, elemente sau sarcini de procesat

T = mulţimea tranziţiilor

I = $P \times T \rightarrow \{ 0 , 1 \}$ este funcţia de intrare specificând arcele orientate de la poziţii la tranziţia de intrare

O = $P \times T \rightarrow \{ 0 , 1 \}$ este funcţia de ieşire specificând arcele orientate de la tranziţii la poziţia de ieşire

m = $P \rightarrow N$ este mulţimea marcajelor accesibile; $\mathbf{m_0}$ este marcajul iniţial

Definiţie: Un marcaj M este o funcţie care alocă fiecărei poziţii ale unei reţele P/T un număr întreg, mai mare sau egal cu zero, de jetoane, reprezentate prin buline negre. Marcajul unei reţele la un moment dat este reprezentat sub forma unui vector coloană M, fiecare componentă a acestuia $m(p_i)$ având valoarea egală cu numărul de jetoane de marcaj în poziţia p_i la momentul respectiv.

Definiţie: O reţea marcată <R, M_0> este o reţea R cu un marcaj iniţial M_0.

O tranziţie $t \in T$ este validată de un marcaj M dacă $\forall p \in P$ avem $m(p) \geq I(p, t)$. Execuţia unei tranziţii t generează un nou marcaj, M'cu $m'(p)=m(p)+O(p, t)-I(p, t)$.

Notaţie: Fie marcajul M care validează tranziţia t_i. Fie M' marcajul la care se ajunge după execuţia tranziţiei t_i. Această evoluţie a reţelei se notează $M[t_i>M']$.

Observaţie: Se spune că un marcaj M' acoperă marcajul M, adică $M' \geq M$, dacă $\forall p_i \in P$, avem $m'(p_i) \geq m(p_i)$.

O reţea marcată $<R, M_0>$ este:

- ***marginită*** (bounded), dacă există un număr natural k astfel încât $M(p) \leq k$ pentru fiecare poziţie p şi pentru fiecare marcaj $M \in A(R, M_0)$;
- ***sigură*** (safe sau 1-marginită), daca $M(p) \leq 1$ pentru fiecare poziţie p şi pentru fiecare marcaj $M \in A(R, M_0)$;
- ***conservativă*** (conservative), dacă există un vector de numere naturale Y astfel încât $Y^T.M=Y^TM_0$ pentru fiecare marcaj $M \in A(R, M_0)$;
- ***viabilă*** (live) dacă pentru fiecare marcaj $M \in A(R, M_0)$ există o secvenţă de execuţie care conţine toate tranziţiile ;
- ***reversibilă*** (reversible), dacă marcajul iniţial M_0 este accesibil din fiecare marcaj accesibil $M \in A(R, M_0)$.

Definiţie: *O poziţie p a unei reţele <R, M_0> este* ***implicită*** *[82] dacă L(R, M_0)=L(N', M_0'), unde <N', M_0'> este reţeaua marcată obţinută prin înlăturarea poziţiei p şi a tuturor arcelor sale de intrare ieşire. O poziţie p a unei reţele R este structural implicită dacă există un marcaj iniţial M_0 astfel încât p este implicită în <R, M_0>.*

Observaţie: *RP conservative sunt practic echivalente ca putere de modelare cu automatele de stare finite*, deoarece numărul de marcaje accesibile dintr-un marcaj iniţial dat (adică numărul 'stărilor' sistemului modelat) este finit. De fapt însă, dat fiind că stările sistemului sunt reprezentate prin marcajele posibile şi nu prin stări distincte, chiar şi în această situaţie *reprezentarea RP este mai eficientă*, talia modelului fiind mai redusă, *la aceeaşi cantitate de informaţie conţinută*.

Procedura de sinteză a unui model de sistem de fabricaţie reprezentare RP elementară cu proprietăţi, viabilitate, mărginire şi reversibilitate intrinseci

Pentru clasa de aplicaţii legate de sisteme de fabricaţie reprezentarea RP elementară (cu posibile extensii la RP temporizate şi RP colorate) foloseşte următoarele noţiuni şi concepte specifice [Zhou,93]:

- **Proprietăţile RP şi sistemele de fabricaţie**
  - ***mărginirea sau siguranţa*** asigură că în sistem nu vor apărea depăşiri ale resurselor (de exemplu: depozit). Siguranţa este cazul special 1-mărginit pentru o poziţie operaţională, prin care se asigură că resursa nu poate fi realocată dacă deja se afla în curs o operaţie tehnologică specifică poziţiei.
  - ***viabilitatea*** asigură absenţa blocajelor în sistem şi prin aceasta asigură că sistemul poate produce.
  - ***reversibilitatea*** asigură comportarea ciclică a sistemului şi are, de asemenea, implicaţii în diagnoza şi revenirea din stări de eroare, asigurând condiţiile pentru ca sistemul să poată fi reiniţializat.

Definiţie: Se defineşte k - PME elementară ca o structură particulară de subreţea PN cu proprietăţi intrinseci de viabilitate, mărginire şi reversibilitate, care modelează folosirea unei singure resurse comune de către un număr k de procese concurente (în paralel).

Definiţie formală:

k - PME = (p_E , D), unde:

$\mathbf{p_E} \in P_B$ resursa comună şi este întotdeauna o poziţie sigură în reţea având $m_0(p_E)= 1$

D este mulţimea perechilor de tranziţii $(t_{aj}, t_{bj})_{j = 1,2,..k}$ unde tranziţiile t_{aj} sunt tranziţii de intrare (lansare proces j din grupul de procese concurente pe resursa p_E), iar tranziţiile t_{bj} sunt tranziţii de ieşire (încheiere p proces j din grupul de procese concurente pe resursa p_E)

Condiţii de verificare structura k-PME

1. (∃) ***k procese distincte*** formate din căi elementare formate din operaţii $p_i \in P_A$, astfel încât grupul (t_{aj} , t_{bj}) defineşte lansarea şi respectiv încheierea unui proces care consumă resursa p_E pe toată durata executării operaţiilor în care este descompus procesul:
   - din punct de vedere topologic resursa comună este cuplată la fiecare dintre procese printr-o singură pereche de arce (unul de intrare, unul de ieşire);
   - orice circuit elementar care include resursa comună p_E şi operaţii de tip P_A trebuie să conţină în secvenţa de descriere topologică tranziţia de încheiere procese;

$$EC(p_i) = p_i \ldots\ldots\ldots \underbrace{t_{ai} \ldots\ldots\ldots t_{bi}}_{P_A} \ldots\ldots\ldots\ldots p_i$$

2. Asocierea/asamblarea între obiecte/servicii de diferite tipuri poate fi modelată numai printr-o tranziţie comună a două procese distincte urmată de o poziţie P_A

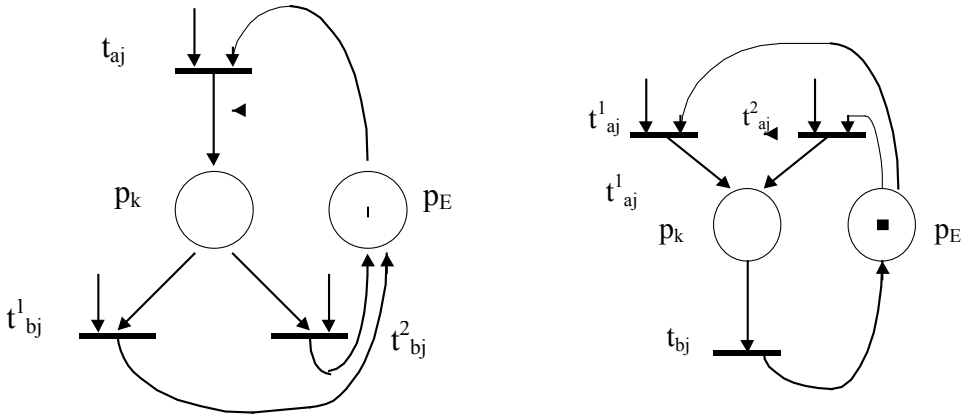
pentru operaţii de asociere. Topologic, pentru graf înseamnă că intersecţia a două căi elementare care modelează fluxuri diferite de obiecte sau obiecte/servicii are loc întotdeauna la nivelul unui nod de tip tranziţie

3. Fiecare dintre procesele concurente (k) are drept egal de acaparare a resursei comune.
4. Odată acaparată resursa p_E de către un proces k, prin lansarea tranziţiei t_{aj}, aceasta va fi în mod necesar eliberată în momentul tranziţiei finale t_{bj}.

Generalizare:

Pentru structura k - GPME (structura k - PME generalizată) se cere verificată structura specifică prin care un proces şi numai unul dintre cele k procese concurente poate fi alimentat de la cel puţin două tranziţii de intrare t^i_{aj} , (t^1_{aj} , t^2_{aj}), (i este numărul tranziţiei de intrare, iar j unul din cele k procese paralele), sau rezultatele obţinute pot fi utilizate de cel puţin 2 căi de ieşire t^i_{bj} , (t^1_{bj} , t^2_{bj}).
Toate condiţiile de validare vor fi verificate similar cu cele de la k - PME

Definiţie: l - SME elementară este o structură particulară de subreţea PN cu proprietăţi intrinseci de viabilitate, mărginire şi reversibilitate, care modelează folosirea unei singure resurse comune de către un număr de procese 1 - PME sau grupuri de procese k - PME / GPME aflate în relaţie de succesiune (grupuri secvenţiale).



Definiţie formală: l - SME = (p_E , $D^1 \cup D^2 \cup \ldots \cup D^l$), unde:

$\mathbf{p_E} \in P_B$ resursa comună şi este întotdeauna o poziţie sigură în reţea având $m_0(p_E) = 1$

$\mathbf{D^j}$ = este mulţimea perechilor de tranziţii $(t^i_{aj}, t^i_{bj})_{i = 1,2,..l}$ unde perechile de tranziţii lansează / închid grupurile de procese secvenţiale

Condiţii de verificare structura l-SME:

- (p_E , D^i) reprezintă structuri de subreţele de excludere mutuală paralel k – PME;
- Între două procese/grupuri de procese paralele relaţiile sunt întotdeauna de secvenţiabilitate astfel încât între două tranziţii şi operaţii D^i şi respectiv D^j (i>j) există o cale de procese A-EP (poziţii P_A);
- nu există nici o cale de procese A-EP care să cupleze pe reacţie inversă grupul D^i cu grupul D^j ;
- orice secvenţă realizabilă care conţine pe t_{bj} şi pe t_{bi}(i>j) trebuie să conţină şi tranziţia t_{ai}, g = ……t_{bj}…..(t_{ai}……t_{bi})….

Generalizare:

Ca şi în cazul proceselor paralele k - GPME există structuri de subreţele l - GSME care au ca specifice topologii de graf RP cu mai multe tranziţii de intrare T_{aj}={ t^1_{aj}, t^2_{aj}} sau de ieşire T_{bj} ={ t^1_{bj}, t^2_{bj}} pentru un grup de procese D^j

Formal ∃ [p_E, $D^1 \cup D^2 \cup \ldots \cup D^l$], l≥2, unde D^i = { T^i_{ai}, T^i_{bi} }

Metodologia de siteză a modelului:

Procedura de sinteză descendentă, de la nivel de activităţi/lucrări la nivel de procese/operaţii tehnologice elementare poate fi sintetizată în următorii trei paşi:

1. Se descompune sistemul în subsisteme independente, corespunzătoare fluxurilor tehnologice distincte din scenariul de fabricaţie propus de proiectant. Pentru fiecare subsistem se alege o subreţea originală care explicitează resursa comună variabilă (P_c) şi lucrările P_{1i}, P_{2i},….,P_{ki} ∈ P_A efectuabile asupra unui lot de piese identice;

2. Prin utilizarea modulelor de proiectare standard (procese serie, procese paralel, procese de decizie cu priorităţi predefinite sau priorităţi liber alese, lucrările se descompun în paşi succesivi până la nivelul de explicitare dorit, astfel încât nu mai sunt necesare explicitări suplimentare procese/operaţii elementare. Prin metoda propusă numărul de stări P_A creşte împreună cu numărul de tranziţii pentru lansarea/încheierea operaţiilor elementare;

3. Pentru fiecare maşină alocată unui proces elementar se adaugă resurse fixe necomune, care modelează starea de disponibilitate a resursei, explicitată printr-un marcaj m_0 ($p'_{i.}$) = 1, astfel încât să creeze structuri 1 - PME. Creşte astfel numărul de P_B alocate sistemului;

Rezolvarea problemelor legate de interacţiuni între subsisteme, generate de utilizarea în comun, de către mai multe procese, a unor resurse comune, roboţi, sisteme de transport etc.

Obiectivul generării unui model complet pentru sistem SDED, care să nu aibă blocaje în timpul funcţionării normale (fără avarii la nivelul utilajelor), se poate obţine prin cuplarea subreţelelor cu ajutorul unor module standard care modelează structuri cu excludere mutuală paralel (k-PME) sau/şi serie (l-SME) şi eventual generalizate (k-GPME) şi (l - GSME).

Integrarea sistematică, în modul, a resurselor comune se poate realiza astfel:

4. Adăugarea resurselor tip depozit (*buffer*). Explicitarea modulului care conţine modelul dispozitivului de depozitare se face conform cu politica de stocare acceptată de către proiectant. Modulul va integra o poziţie $p_j \in P_A$ pentru modelarea operaţiilor de stocare cu o structură de poziţii tip P_B ajutătoare;
5. Se adaugă resursele partajate tip PME şi GPMS;
6. Se adaugă resursele partajate tip SME şi GSME şi se stabileşte marcajul iniţial.

Metoda a fost utilizată pentru dezvoltarea modelelor sistemului pilot de fabricaţie din L1 (cap.5).

3.2.4. DIMENSIUNEA - REPREZENTARE CONTEXTUALĂ

Pentru a implementa un sistem software de control, în concordanţă cu standadardele de achiziţie de date şi supervizare pentru o arhitectură de sistem deschis în fabricaţia discretă se vor utiliza produse software ce se vor adapta la cerinţele de control al proceselor, se vor crea interfeţe specifice om-maşină şi structuri de date necesare. Task-urile specifice vor fi realizate, într-o arhitectură de control distribuit şi achiziţie de date prin intermediul comunicaţiilor om-maşină, achiziţie de date, procesare de date, mentenanţă de sistem.
Se vor integra modele de tip:

- modele de timp real;
- modele de diagnoză/prognoză;
- modele de ordonantare.

Din figura 4.1. referitoare la globalizarea fabricaţiei se poate observa că modelele „fizice” obţinute în mediul virtual de proiectare şi cele fizice obţinute prin prototipizare rapidă se constituie într-o reactualizare a rolului modelelor „fizice” şi a tehnologiilor de realizare a lor.

Suportul informatic şi dezvoltarea sistemelor senzoriale integrate au creat cadrul implicării beneficiarilor direct în fazele de proiectare şi reproiectare a produsului în mediul „*full virtual*” pentru prototipizare rapidă. De asemenea, modelele de la

nivelul fabricii virtuale, asociate facilităţilor din fabrica fizică permit simularea proceselor de fabricaţie, în care facilităţile nu sunt considerate numai nişte blocuri funcţionale animate, având anumiţi parametri programabili (SIMAN, CINEMA, Taylor II) timp de procesare, durate de încărcare etc., ci sunt modele complexe ale maşinii, care permit urmărirea ciclului de procesare, deplasarea elementelor componente, urmărirea în mediu virtual a întregii evoluţii la nivel de maşină. Aceste modele virtuale ale maşinilor din aplicaţiile de la nivelul fabricii fizice sunt denumite ***„modele de înaltă fidelitate ale echipamentelor”***.

Pentru a dezvolta astfel de modele sunt utilizare aplicaţii de ***NC-uri virtuale***, care au ca scop studiul efectelor obţinute asupra ciclului maşină, prin diferite secvenţe de amplasare a componentelor, sau diferite configuraţii de iniţializare a echipamentelor [Bodner,97].

Aceste modele de înaltă fidelitate a maşinilor sunt utilizate în prima etapă ca mijloace în analiza de tip ***„ce se întâmplă dacă (what-if)”.***

Def. Modelul virtual al maşinii *este un model extrem de detaliat al acesteia, utilizat pentru programarea off-line şi analiza operaţională. El se caracterizează print-o reprezentare tehnic precisă a structurii şi comportării maşinii şi printr-o animaţie realistă a maşinii.*

În general, dezvoltarea unui astfel de model constă în două etape majore:

- stabilirea unui model animat, de înaltă fidelitate, a maşinii;
- caracterizarea duratei ciclului de prelucrare pe maşină, inclusiv testarea diferitelor configuraţii ale maşinii [Grotzinger,92, Wilhelm,92].

Modelul trebuie să capteze comportarea statică şi dinamică a maşinii fizice precum şi interacţiunile acesteia cu alte mecanisme din sistem (roboţi de alimentare, mecanisme de transport, testare, calibrare etc.)

Principala dificultate constă în agregarea datelor referitoare la rularea modelului. Elementele de geometrie ale maşinii precum şi cele asociate produsului (în exemplul de faţă, circuitul imprimat) trebuie stabilite cu precizie, iar duratele de mişcare a diferitelor elemente trebuie, de asemenea, precizate, deşi ele variază şi nu sunt uşor de măsurat (se ţine cont de tipul mişcărilor, profilele de accelerare-decelerare, vitezele maxime admise, interblocări etc.).

Experienţa din robotică, cu rezultatele obţinute în calibrarea automată, sistemele de învăţare a traiectoriei, aplicaţiile *off-line* de programare asigură că astfel de abordări conduc la rezultate. Tehnicile de identificare utilizate în robotică pot fi aplicate selectiv la identificarea parametrilor maşinilor.

Ca şi în robotică stabilirea unui model de înaltă fidelitate are cel puţin două obiective:

- stabilirea parametrilor dinamici necesari în algoritmi evoluaţi de conducere;
- programarea-simularea *off-line* a aplicaţiilor robotizate.

Dacă primul obiectiv este unul legat de dezvoltatorul (fabricantul) de sistem mecatronic, fiind o problemă de „*embeded system*“, pentru al doilea obiectiv se pot utiliza medii existente, în care particularizarea modelului sistemului mecatronic (parametrii statici şi dinamici) poate fi făcută în faza de construire a modelului, utilizând parametri oferiţi de fabricant sau cei obţinuţi prin identificare.

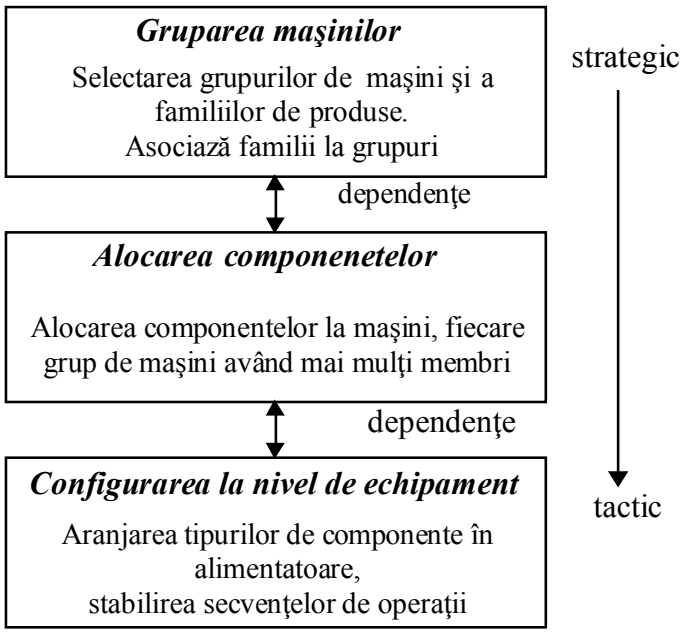


Fig. 3.13.

Există o ierarhie a problemelor de decizie care trebuie rezolvate pentru a putea dezvolta un plan de procese şi o configuraţie a echipamentelor necesară producerii unei game extinse de subansamble conform figurii 3.13 [McGinnis,92], iar modelele de înaltă fidelitate a maşinilor şi facilităţilor implicate se constituie ca suport pentru analiza diverselor soluţii posibile.

În afară de modelare, simularea este un instrument important în studiul sistemelor de fabricaţie şi în sinteza politicilor de conducere pentru acestea.

Prin simulare se încearcă găsirea unei soluţii cu maximum de robusteţe faţă de un context slab definit. Simularea este un instrument de analiză, iar realizarea unei aplicaţii de simulare presupune următorii paşi [Banks,96]:

- ***definirea problemei***: definirea sitemului de analizat şi problemele la care se asteaptă răspuns prin simulare;
- ***construirea modelului;***
- ***achiziţie de date:*** colectează datele necesare pentru o specificare completă a modelului: valorile parametrilor, valorile iniţiale ale parametrilor etc.;

- ***verificarea şi validarea modelului:*** utilizarea modelului şi validarea acestuia, astfel încât comportarea lui să reflecte comportarea sistemului real pe care îl modelează, cu gradul de precizie stabilit;
- ***experimente:*** stabileşte modificările necesare modelului şi repetă pasul anterior până la obţinerea performanţelor impuse;
- ***evaluarea rezultatelor:*** se analizează rezultatele şi se verifică dacă acestea corespund specificaţiilor impuse şi dacă nu, se reiau experimentele;
- ***optimizarea modelului;***
- ***intrepretarea rezultatelor.***

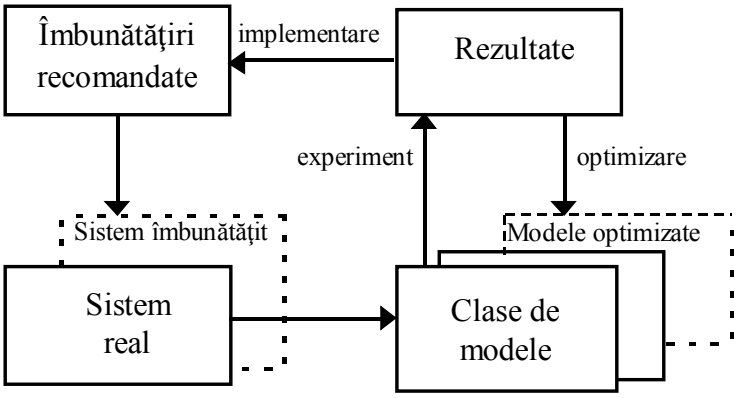


Fig. 3.14.

În ***fabricator***, nivelul de ***fabrică virtuală*** capătă semnificaţia unui mediu de modelare, simulare şi validare aplicaţii *off-line* din punctul de vedere clasic al înţelesului, dar *on-line* din perspectiva unui mediu de prototipizare „*full virtual*". Acest mediu este construit din ***modele virtuale ale maşinilor*** acţionate de ***NC virtuale*** realizează produse create în mediu de inginerie concurentă pentru care procesele de prelucrare sunt stabilite prin simulare, iar programele pentru ***open CNC-uri în*** cod standard, şi se poate constitui ca un model de timp real al întregii aplicaţii ***fabrica fizică,*** care să contribuie la validarea prin simulare de noi soluţii la nivelul funcţionării fabricii fizice şi să asigure, de asemenea, suportul de decizie la nivelul MCS. Programele pentru open CNC-uri, roboţi, AGV-uri sunt validate în aplicaţiile virtuale.

Afirmaţiile sunt susţinute atât de experienţa anterioară din mediile de modelare, simulare şi programare *off-line* pentru roboţi industriali, cât şi de cele mai noi aplicaţii de modele virtuale ale maşinilor din industria electronică.

Sinergia modelelor

Modelele de produse şi procese precum şi modelele de resurse sunt diverse şi redundante. Ordonanţarea utilizează un model de facilităţi care nu face apel la funcţiile de fabricaţie ale acestora dar utilizează durate de prelucrare, capacitatea, cicluri de mentenanţă şi reparare precum şi alte proprietăţi relevante pentru a putea alcătui planul de ordonantare sau re-ordonantare. Agentul de monitorizare presupune achiziţia şi evaluarea informaţiilor legate de funcţionarea echipamemtului şi proprietăţile sale de detaliu, posibil de evidenţiat prin informaţia senzorială. Agentul de monitorizare face apel la informaţia utilă ordonantării numai pentru a putea stabili corelaţii între defecte ale produselor şi proprietăţi ale facilităţilor. Modelele se suprapun între ele, iar agentul alege modelul potrivit pentru task-ul pe care trebuie să-l implementeze.

„Operatorii" umani au responsabilităţi bine stabilite în sistemul de fabricaţie, cu accent pe decizie. Ca agenţi inteligenţi, ei interacţionează cu alţi agenţi care îi sprijină în obţinerea informaţiilor (filtrate, convertite, combinate, sumarizate) de suport pentru adoptarea deciziei. VF pune accent pe reutilizarea agenţilor/obiectelor.

Se poate preciza că în etapa curentă nu se pune problema interoperabilităţii între modele, scopul esenţial al cadrului de multimodelare 4D fiind acela al sublinierii sinergiei dintre diversele clase de modele. Din cauza lipsei standardelor în domeniu problema interoperabilităţii ramâne o problemă de context, cu soluţii particulare (ex. IDEF$\Leftrightarrow$CPN, PN$\rightarrow$interpretor$\rightarrow$Program executabil). Utilizarea unei platforme de integrare activităţi de tipul de tipul celei care va fi prezentată în cadrul studiului de caz, CAS-CADE se poate constitui într-o „infrastructură" care să asigure funcţiile de bază şi care să constituie suportul pentru dezvoltarea unui mecanism *plug-and-play software*, prin care diverse clase de modele să schimbe structuri de date prin care să se asigure interoprabilitatea. Acest domeniu se poate constitui într-o direcţie de cercetare viitoare.

CONCLUZII

*Termenul de **fabrică virtuală** a fost utilizat iniţial pentru a identifica o **"fabrică agilă"** şi face referire la un mediu virtual în care sistemele software pentru controlul cu reacţie al proceselor de fabricaţie şi diferitele scenarii de acţiune pot fi testate şi executate fară a utiliza direct echipamentele fizice.*

***Fabrica virtuală** este un cadru de multimodelare care asigură platforma de integrare a controlului avansat cu funcţiile de afaceri ale sistemului de fabricaţie, şi oferă cadrul necesar dezvoltării întreprinderii, pe baza unui model cu învăţare, în conformitate cu o strategie de dezvoltare incrementală. Abordarea orientată platforma de multimodelare pentru suport decizie, dezvoltare şi operare sistem de fabricaţie manifestă o tendinţă convergentă cu topica curentă de cercetare în direcţia Integrării în Fabricaţie (IiM). Modulele software de control interacţionează cu o suită de module software care simulează atât funcţionarea echipamentelor cât şi procesele de fabricaţie.*

Se propune o reprezentare a contextului dezvoltării conceptului de fabrică virtuală integrat ansamblului paradigmelor generate de evoluţia suportului informatic şi al mediilor electronice de comunicaţii, inginerie concurentă, re-ingineria proceselor de afaceri, comerţul electronic, întreprindere virtuală.

Definiţia propusă pentru fabrica virtuală subliniază că aceasta este un sistem de modele şi aplicaţii software care descriu în mediu virtual evoluţia dinamică a unui sistem de fabricaţie real, cu scopul rezolvării unor clase specifice de probleme: asistarea adoptării deciziei strategice/tactice/operaţionale, dezvoltarea de produse/procese/facilităţi de fabricaţie, dezvoltarea sistemului automat de control, managementul de procese, alocarea resurselor.

Se prezintă arhitectura fabricii virtuale constând din patru subsisteme: subsistemul management afaceri, subsistemul modelare produse/procese, subsistemul facilităţi, subsistemul management de procese. În conformitate cu cerinţele pentru sistemul post-CIM care subliniază nevoia de autonomie şi co-operare este unanim acceptată aserţiunea ,,resursele de fabricaţie actionează ca agenţi care generează procese având ca obiectiv satisfacerea unei structuri scop" furnizată de ,,motorul" de decizie orientat după reguli de afaceri, pentru a asigura livrarea produselor în conformitate cu un model de date produs (PDM- Product Data Model), dezvoltat în mediul de inginerie concurentă (CE-Concurrent Engineering).

Conceptele utilizate sunt: structura scop, procese de fabricaţie, structura de procese, situaţii-operaţii, protocol de comunicaţii între scena externă şi cea internă , agenţi autonomi, istoria unui agent.

Pentru a modela fabrica virtuală se propun tehnici de programare orientate spre obiecte, iar agentul se defineşte ca obiect cu scop. Este tratată problema descompunerii unei funcţii în sub-funcţii şi alocarea acestora unor agenţi.

Modelarea alocării funcţiilor ţine cont de încărcarea fiecărui agent precum şi de efortul necesar coordonării interacţiunii dintre aceştia.

Se utilizează descompunerea în scopuri-obiective-activităţi (frecvent utilizată când se face referire la agenţi umani) sau în termeni de misiuni-funcţii-activitaţi. Se utilizează un model de execuţie a activităţii de către o singură resursă, care utilizează cinci niveluri pentru a defini rolurile: evaluare situaţii, fuziune informaţii, procesare activităţi, interpretare comandă, selectare răspuns. Se introduce noţiunea de fabricator, pentru a desemna formal un sistem de fabricaţie, în condiţiile determinate de apariţia conceptului de fabrică virtuală.

Definiţia propusă pentru fabricator subliniază că acesta este un agent autonom care oferă produse şi servicii pieţei de afaceri. El constă din trei subsisteme ierarhice: subsistemul fabrica fizică, constând în totalitatea echipamentelor programabile din sistemul de fabricaţie, subsistemul fabrica virtuală şi subsistemul claselor de sisteme de automatizare prin care fabrica virtuală interacţionează cu fabrica fizică.

Definiţia formală propusă pentru fabricator subliniază faptul că este un meta-sistem deschis aflat într-un univers de afaceri, constând dintr-o mulţime de resurse organizate în fabrica fizică, un model de fabrică virtuală, mulţimea claselor de sisteme de automatizare prin care fabrica virtuală interacţionează cu fabrica fizică, şi o structură scop.

Se propune un cadru de modelare 4D orientat: funcţional, structural, comportamental şi contextual pentru analiza, proiectarea şi implemenatrea unui sistem de control al unui sistem de fabricaţie, ca nucleu de cercetare privind sinergia diverselor clase de modele şi a cerinţelor de interoperabilitate a acestora.

Din considerente de calitate a proiectării sistemului automat se propune o metodologie de proiectare colaborativă în spirală, astfel încât analiza, proiectarea şi reproiectarea să se poată face prin iteraţii succesive, la diverse niveluri de rafinare a modelului.

Semnificaţia proiectării în spirală poate fi asociată şi cu un proces de micşorare a incertitudinii pe măsura parcurgerii spiralei între cele patru dimensiuni şi prin coborâre în ierarhie. Pentru fiecare dintre dimensiuni se propun clase de modele dintre care se analizează: metodologia Yourdon pentru dimensiunea funcţional, modelul CIM orientat patternuri pentru dimensiunea structural, formalismul reţelelor Petri pentru dimensiunea comportamental şi se prezintă modelele maşinilor virtuale pentru dimensiunea contextual.

Se precizează că în etapa curentă nu se pune problema interoperabilităţii între modele, scopul esenţial al cadrului de multimodelare 4D fiind acela al sublinierii sinergiei dintre diversele clase de modele; astfel, datorită lipsei standardelor în domeniu problema interoprabilităţii rămâne o problemă de context, cu soluţii particulare(de exemplu: IDEF⇔CPN, PN→interpretor→Program executabil).

Utilizarea unei platforme de integrare activităţi de tipul celei ce va fi prezentat în cadrul studiului de caz, CAS-CADE se poate constitui într-o ,, infrastructură" care să asigure funcţiunile de bază, într-un mediu OO, prin care diverse clase de modele să schimbe structuri de date prin care să se asigure interoperabilitatea. Acest domeniu se poate constitui într-o direcţie viitoare de cercetare.

Capitolul 4 – cuprins

4.1. Sistemul întreprindere virtuală **113**

4.1.1. Arhitectură de referinţă pentru întreprinderea virtuală **115**

4.1.2.Interfaţă neutrală pentru integrarea partenerilor în întreprinderea virtuală **119**

4.2. Proiectarea de alianţe tip întreprindere virtuală, bazată pe module fabricator **128**

4.2.1. Reţea globală de cunoştinţe de reinginerie **128**

4.2.2. Suport formal pentru proiectarea de alianţe tip întreprindere virtuală **130**

- **Definire operatori specifici** **131**
- **Algoritm de proiectare a întreprinderii virtuale bazat pe module fabricator** **135**

4.2.3. Bibliotecă cu acces inteligent la cunoştinţe de reinginerie **139**

4.3. Problema conducerii sistemelor de fabricaţie integrate în structuri de întreprindere virtuală **143**

4.4. Arhitectură de control supervizor şi achiziţie de date pentru fabricator **148**

Concluzii **152**

4

SISTEMUL ÎNTREPRINDERE VIRTUALĂ

4.1. SISTEMUL ÎNTREPRINDERE VIRTUALĂ

Întreprinderea virtuală este un sistem distribuit geografic în care se regăsesc următoarele funcţiuni [Oliveira,97]:

- recepţionare ordine şi răspunsul rapid la acestea;
- configurarea structurii virtuale pe baza comunicării în reţea;
- sistem de planificare global;
- sistem de planificare şi ordonanţare local;
- control proactiv la nivel de VE;
- control reactiv la nivel de partener în alianţa VE, fabricator.

Aceste funcţiuni asigură:

- verificarea în timp real a ordinelor, din punct de vedere al posibilităţilor de realizare (fezabilitate) şi a duratelor de livrare;
- configurarea VE prin negociere şi stabilirea/verificarea capacităţii de a livra produse la termenele prevăzute prin contract;
- stabilirea ordinii activităţilor necesare satisfacerii comenzii, optimizarea consumului şi rutelor, maximizarea profitului;
- achiziţia şi prelucarea de date necesare monitorizării stării comenzilor pentru a evita întârzieri în livrare;
- controlul la nivel de VE şi local, la nivel de fabricator, având ca scop menţinerea sistemului „alianţă virtuală" în zona normală de funcţionare;
- schimbul de informaţii necesare organizaţiei virtuale, care să sprijine totalitatea funcţiunilor prevăzute;
- interfeţe standard către alte aplicaţii cu care VE se interconectează.

Mediul în schimbare al globalizării afacerilor, figura 4.1., deschide o cale către noi domenii ale schimbului de informaţii în spaţiul electronic şi pentru comerţul electronic, în care partenerii alianţelor virtuale trebuie să acţioneze integrându-se în lanţuri de companii organizate ca lanţuri logistice care schimbă informaţii esenţiale în scopul menţinerii competitivităţii şi a oportunităţii de afaceri.

Definiţia 1. Întreprinderea virtuală *este o alianţă temporară de întreprinderi care se asociază pentru a utiliza în comun (partaja) experienţă şi resurse cu scopul de a răspunde mai bine la cerinţele de afaceri.*

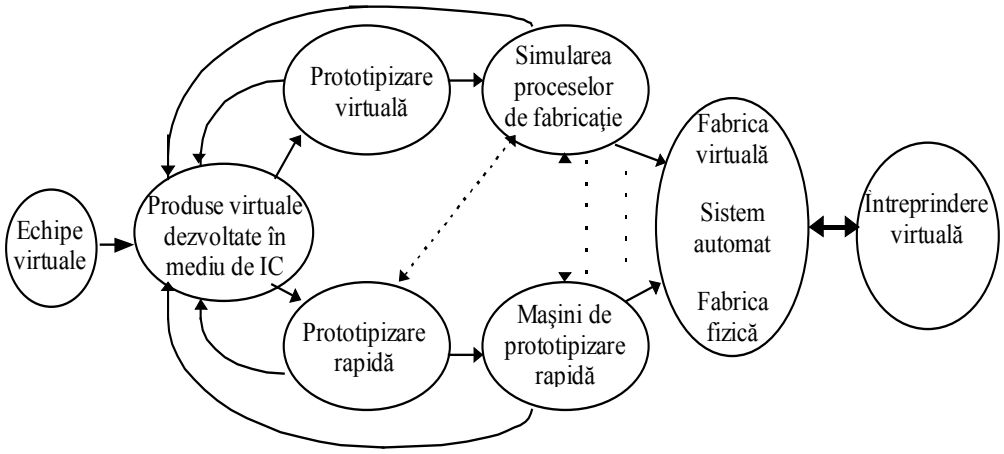


Fig. 4.1.

Definiţia 2. Întreprinderea *virtuală este o alianţă temporară de parteneri focalizaţi pe* ***competenţele lor de bază, capabilă să asigure cooperarea****, ca suport al procesului de* ***inovaţie în reţea*** *şi pentru a răspunde rapid la cerinţele de afaceri.*

Trasături esenţiale ale VE sunt cooperarea partenerilor focalizaţi pe competenţele lor de bază şi inovaţia în reţea, care se caracterizează prin :

- ***capacitatea de inovare*** (în cultura organizaţiei, forţa de inovaţie – forţa financiară, strategie, cunoştinţe despre piaţă, inovaţie în management, management proiecte, etc.);
- ***capacitatea de procesare*** (organizare dinamică – structuri dinamice, managementul fluxurilor de lucru şi sisteme noi de informaţii, competenţa şi posibilităţile tehnologice);
- ***capacitatea de cooperare*** (abilitatea de lucru în echipă, înlăturarea barierelor în mentalitate privind cooperarea, gândirea în reţea, etc.).

Alianţa este sprijinită de suportul informatic (tehnologiile de comunicaţie şi reţelele de calculatoare).

Interconectarea partenerilor într-o alianţă - întreprindere virtuală, ridică următoarele probleme:

- infrastructura de comunicaţii şi aspectele legate de siguranţa informaţiei;
- managementul interacţiunilor, bazat pe standardizarea schimbului de informaţii;

- EDI – pentru schimbul de date administartive şi EDIFACT pentru comenzile clienţilor şi facturile către aceştia;
- STEP pentru schimbul de date tehnice (de exemplu: CAD/CAPP);
- managementul fluxurilor (*workflow*);
- selectarea partenerilor şi configurarea/re-configurarea alianţei VE.

Definiţia 3. Întreprinderea virtuală *poate fi formal definită ca un meta-sistem deschis, aflat într-un univers de afaceri în care evoluează o mulţime de* ***Fabricatori****, care utilizează reprezentări multiaspect ale produselor/proceselor şi cooperează în cadrul unor alianţe temporare multi-fabricator pentru a-şi atinge setul de obiective :*

VE = (U, F, M v, G) unde:

U *= un univers de afaceri*
F *= setul de fabricatori*
M v *= o reprezentare multiaspect a produselor/proceselor*
G *= o structură de obiective (orientare-afaceri)*

Configurarea prin negociere a VE urmează două macro-faze:

- ***faza de configurare preliminară a alianţei***, pentru participare la BID, când alianţa este virtuală manifestându-se numai prin obiectivul major al câştigării competiţiei; VE participă la competiţie ca ***jucător P*** care capturează scopul de afaceri constând în câştigarea competiţiei în spaţiul virtual de afaceri; după câştigarea acesteia, jucătorul P devine ***jucător global PG***, care capturează scopul VE în conformitate cu o imagine a strategiei generată de către echipa de conducere la vârf a alianţei VE pe durata dezvoltării structurii reţelei de afaceri (de exemplu: proiectarea şi ingineria de service, ofertantul de componente, distribuitori, producţie, negociator-broker), concretizată în lanţurile de alimentare create;
- ***faza de configurare a alianţei VE operaţionale***, care urmează momentului câştigării BID şi are ca scop rafinarea structurii scop orientată agent sau obiectiv (VF) şi stabilirea structurii operaţionale a VE, capabile să ofere serviciile contractate, la momentele stabilite şi în condiţiile de costuri impuse. ***Jucătorul global PG***, pe baza analizei de performanţă a VE decide asupra necesităţii reconfigurării alianţei.

4.1.1. ARHITECTURA DE REFERINŢĂ PENTRU ÎNTREPRINDEREA VIRTUALĂ

Studiul întreprinderii virtuale (VE) impune definirea unei arhitecturi de referinţă, proiectarea şi dezvoltarea unei platforme, utilizarea unor protocoale de

comunicaţii şi a unor mecanisme care să garanteze siguranta transmiterii informaţiei.

Într-o VF componenetele actionează ca noduri într-o reţea de furnizori, clienţi, inginerie, şi furnizori de alte servicii specializate. Întreprinderea virtuală se "materializează" prin selectarea competenţelor şi a mijloacelor (resurselor) din diferite noduri (firme) şi integrarea lor aparent într-o singură "entitate de afaceri".

O arhitectură posibilă a organizaţiei virtuale este arhitectura scalabilă, conform figurii 4.2.

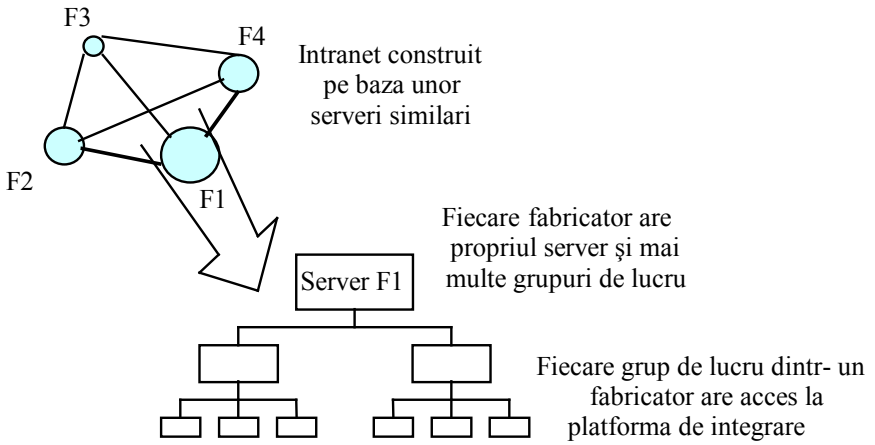


Fig. 4.2. Arhitectura de organizaţie virtuală

Interfaţa grafică utilizator (GUI) pentru VE prezintă şi un sistem de referinţă spaţial care permite echipelor ditribuite geografic să lucreze ca şi cum ar fi în acelaşi loc. Un scenariu operaţional posibil este cel de conferinţă desktop, ca în figura 4.3.

Referitor la VE trebuie facută precizarea că există numeroşi termeni, întreprindere extinsă, managementul lanţurilor logistice de alimentare, comerţ electronic, care reprezintă însă concepte asociate sau vederi parţiale ale întreprinderii virtuale.

Întreprinderea virtuală poate fi clasificată în conformitate cu un număr de caracteristici :

- ***DURATA***. Există alianţe care durează pe parcursul ,,unei singure oportunităţi de afacere". Acest tip de alianţă este cel mai frecvent întâlnită şi poate fi exemplificat printr-un parteneriat creat pentru a construi un obiectiv industrial. Lanţurile de alimentare din industria de autoturisme sau aeronautică sunt alianţe ,,pe termen lung", care acoperă un număr indefinit de procese de afaceri.

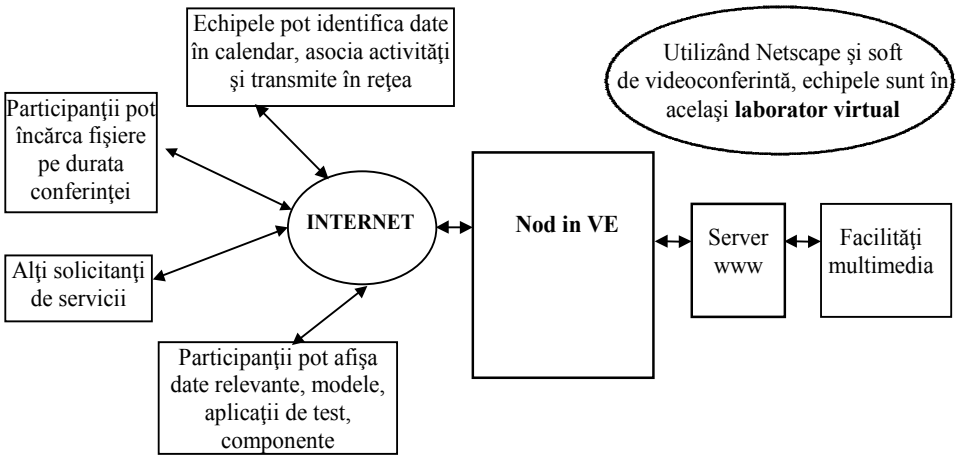


Fig. 4.3. Conferinţa desktop

- ***TOPOLOGIE***. Clasificarea se face ţinând cont de topologia reţelei de noduri create. Reţeaua de noduri din VE este o structură variabilă/dinamică în care partenerii pot să se asocieze dinamic şi să părăsească alianţa odată cu încheierea procesului de afaceri care a determinat alianţa (uzual parteneri ne-strategici). Parteneriatele strategice determină o structură relativ fixă a reţelei. Alt aspect important al topologiei este acela că partenerii pot face parte simultan din mai multe reţele (VE) sau pot fi asociaţi uneia singure.
- ***COORDONAREA.*** Există un număr variat de modele relativ la coordonarea reţelei de parteneri. Pentru domeniul industriei de automobile şi avionică reţeaua este de tip stea, în care întreprinderea care realizează produsul final este înconjurată de reţeaua de furnizori ("alianţe pe termen lung"). În literatura de specialitate se face referire la "regulile jocului" care sunt impuse de ***compania dominantă***. Acestui tip de reţea i-a fost asociat conceptul de **Întreprindere Extinsă**. Există şi ***alianţe democratice*** în care nodurile cooperează pe baze egale, îşi păstrează total autonomia şi pun în comun competenţele lor de bază.
- ***VIZIBILITATEA.*** Se referă la nodurile cu care este interconectat un nod din reţea, respectiv „cât de departe poate vedea" un nod.

Cerinţa generală pentru o VE este că infrastructura informatică de suport trebuie să asigure interoperabilitatea şi schimbul de informaţii, în timp–real, astfel încât organizaţia virtuală să poată funcţiona ca o singură unitate integrată, partenerii păstrându-si totuşi autonomia.

Fiecare partener–aliat dintr-o VE utilizează un sistem propriu de management al informaţiei şi strategii de control specifice astfel încât sistemele proprii de planificare a producţiei şi control (PP&C) trebuie interconectate şi „compatibilizate" electronic.

Contribuţia CIM şi post-CIM în domeniul platformelor funcţionale CAD/CAM, CAE, PP&C etc. se constituie într-o experienţă şi a condus la rezultate în interiorul întreprinderii, care trebuie utilizate şi cărora trebuie să li se aducă "completări" referitor la cerinţele legate de comunicaţiile inter-aplicaţii, în VE. Efortul de cercetare se concentrează asupra separării funcţiunilor interne de cele referitoare la integrarea în reţea, care să asigure integrarea şi sistemelor deja existente şi asupra definirii unor arhitecturi de planificare şi ordonantare în întreprinderea virtuală.

Cerinţele funcţionale pentru operarea VE sunt, conform [Camarinha,97] :

- ***Funcţiuni de bază asociate manipulării informaţiei***:
  - schimbul de informaţii tehnice (modele de produs, rapoarte privind calitatea produselor, etc.) şi informaţii referitoare la afacere (comenzi)
  - videoconferinţe şi cursuri la distanţă;
  - accesul comun la informaţie: documentaţii, cataloage, informaţii de piaţă, specific companie, etc.;
  - informaţii de transmis: noutăţi ştiinţifice, noutăţi de piaţă, oferte;
  - confidenţialitatea datelor;
  - manipularea standardelor EDIFACT (pentru afaceri) şi STEP (pentru date produs).
- ***Funcţiuni legate de materiale:***
  - Logistica;
  - managementul fluxurilor de materiale;
  - previziune nevoi de materiale;
  - utilizarea informaţiei specifice fluxului de materiale.
- ***Funcţia de creare reţea şi configurare:***
  - căutare şi selecţie parteneri;
  - negociere şi managementul contractelor;
  - atribuire roluri şi responsabilităţi partenerilor;
  - stabilirea fluxurilor de informaţii.
- ***Funcţii de acces la servicii electronice:***
  - comerţ electronic;
  - mecanisme de asigurare a securităţii plăţilor;
  - servicii specializate de consultanţă.
- ***Funcţii de Coordonare:***
  - coordonarea la nivel de nod (managementul resurselor, ordonantarea);

- coordonarea în reţea (management resurse şi ordonanţare pentru sistem distribuit);
- Inginerie Colaborativă.

Suportul computerului pentru lucrul în colaborare (**CSCW**-Computer Support Colaborative Work) are rolul de a dizolva graniţele dintre grupurile de lucru prin suportul oferit activităţilor concurente. Activitatea este multi–disciplinară prin natura ei necesitând contribuţii din domeniul calculatoarelor, proiectarea interfeţelor utilizator, psihologie cognitivă, economie, etnologie, sociologie etc. [AIT]. Majoritatea acestor discipline sunt interesate cum lucrează grupurile şi mai puţin suportul tehnologiei informaţiei.

Termenul de CSCW este uzual diferenţiat de „grup de lucru" utilizat curent cu referire la tehnologie şi nu la specificul activităţii [AIT]. Grupurile de lucru sunt caracterizate de factori care influenţează cooperarea: **structură** (grupuri slab sau puternic cuplate), omogenitate (utilizează cunoştinţe şi deprinderi similare) sau eterogenitate, **poziţie** în ierarhie, în cazul structurilor ierarhice, **mărime**.

Aplicaţiile de CSCW includ: desktop şi videoconferinţe, camere de lucru electronic, sisteme de suport grup, comunicaţii electronice. Se poate face referire la numeroase aspecte ale CSCW incluzând diferenţieri temporale şi spaţiale şi caracteristici ale grupurilor care interacţionează.

Condiţiile de timp real pentru anumite activitati (*taskuri)* determină interacţiuni care se deplasează de-a lungul axelor diferitelor dimensiuni. Dimensiunea atemporală se referă la tipul sincron sau asincron al comunicaţiilor; videoconferinţele (sincron), e-mail (asincron).

Distribuţia spaţială a grupului crează situaţii care solicită rezolvarea–separarea geografică şi distanţele mari determină constrângeri temporale.

Predictibilitatea conexiunilor este de asemenea un factor de dificultate, în dimensiunile timp şi spaţiu.

Alt factor este legat de caracteristicile grupurilor care cooperează (structură, omogenitate, poziţie ierarhică, mărime).

4.1.2. INTERFAŢĂ NEUTRALĂ PENTRU INTEGRARE PARTENERILOR ÎN ÎNTREPRINDEREA VIRTUALĂ

Pentru integrarea partenerilor în VE se propune crearea unei interfeţe neutrale, denumite Driver de Integrare (DI). Arhitectura propusă se bazează pe două module:

- ***Modulul Specific Local;***
- ***Driverul de Integrare.***

Modulul Specific Local

Se referă la toate elementele specifice fiecărui laborator (companie) în parte (baza de date proprie, sistemul de management informatic, toate procesele locale de adoptare decizii/activităţi (PPC şi sistemul de inginerie).

Din punctul de vedere al VE, PPC prin funcţiunile sale principale – ***managementul logisticii de întreprindere*** (managementul fluxurilor de comenzi

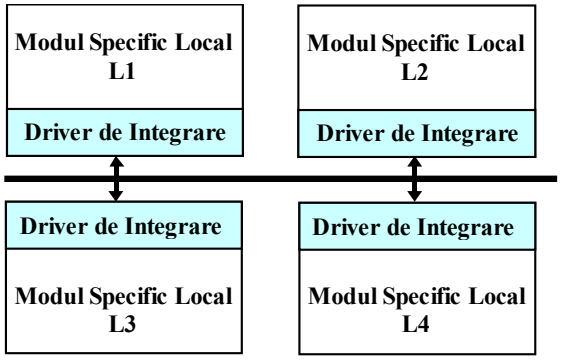


Fig. 4.4. Driver de integrare în fabrica virtuală

managementul datelor produs, planificare vânzări-cumpărări, planificarea contractelor curente), ***ordonanţarea*** principală a productiei, ***controlul*** producţiei, ***controlul/urmărirea*** calităţii, ***costuri*** industriale se constituie în componenţa principală implicată în interacţiunile cu lumea exterioară.

În urma unei analize preliminare [Afsarmanesh,97] informaţiile necesare la nivelul VE sunt cele de ***management al comenzilor*** şi ***management al informaţiilor despre calitatea produselor***.

Managementul comenzilor; fiecare comandă va fi însoţită de o diagramă electronică a tranziţiilor de stare, care să se constituie în istoria comenzii. Pentru a asigura controlul evoluţiei comenzilor fiecare partener trebuie să poată notifica ce parteneri asupra schimbărilor survenite în ciclul de viaţă al comenzii. Tehnologia informaţiei face posibilă asocierea la o comandă a specificaţiilor tehnice pentru produs (de exemplu: lista de materiale, modele geometrice etc.), eventual în reprezentare standard STEP (standard ISO, 1992 pentru schimb de date tehnice). În VE comenzile sunt într-o permanentă dinamică. Modelul propus face apel la noţiunea de ,,*comenzi incomplet formulate"* sau comenzi ,,*imprecis formulate*" (o comandă în care valorile unor atribute sunt specificate într-o formă vagă, care urmează să primească o valoare specificată ulterior; cantitatea, culoarea pot fi atribute vagi).

Comenzile ,,originale" sunt completate odată cu venirea unei ,,noi" comenzi, în care se fac precizările corespunzătoare. Mecanismul va fi asigurat de un subset al

mesajelor EDIFACT şi eventual o extensie a acestuia. Monitorizarea stării comenzilor, eventualele întârzieri în procesarea acestora, incapacitatea temporară a unui partener din lanţul de alimentare, schimbări în comenzi neîncheiate încă, schimbări în eşalonarea livrărilor sunt parametrii care vor putea fi schimbaţi prin intermediul standardelor EDI. Se au în vedere două aspecte: identificarea subsetului de mesaje EDIFACT şi implementarea lor în module EDI şi adaptarea PPC pentru a putea utiliza/furniza informaţiile solicitate din exterior.

Managementul informaţiilor referitoare la calitatea produselor; La nivelul PPC al companiei sunt incluse sau vor fi incluse funcţiunile care să asigure managementul informaţiilor necesare pentru standardul ISO 9000. Pentru VE dificultăţile sunt foarte mari neexistând încă standarde. Cerinţa de garantare a unui nivel de calitate prestabilit pentru subansamblele/componentele incluse într-un produs stabileşte nevoia de schimb de informaţii de acest tip.

Se poate formula un set de cerinţe adaptabil odată cu standardizarea domeniului:

- formatul rapoartelor nu se impune;
- informaţia privind calitatea unui produs nu este implicit disponibilă ci se obţine la cerere;
- pentru fiecare produs trebuie să existe o istorie a sa, care să îl însoţească şi care să fie completată de-a lungul întregului ciclu de viaţă al procesării acestuia; fiecare modul adaugă responsabilitatea sa pentru „valoarea nou creată”;
- la nivelul fiecărui nod se „urmăreşte” calitatea subansamblelor utilizate de întreprindere dar furnizate de alte noduri din reţea.

Driverul de Integrare (DI)

Conţine toate funcţiile necesare pentru interconectare între laboratoare. Elementele propuse pentru a face parte din driverul de integrare sunt [Camarinha,97] :

1. ***Modulul de management informaţii:*** modelează şi asigură managementul informaţiilor de suport al cooperării

2. Modulul de coordonare locală (MCL):

- execută/controlează planul de fluxuri de activităţi;
- asigură comportarea DI şi interacţiunea sa cu alte module;
- tratează evenimentele privind colaborarea în concordanţă cu regulile stabilite; evenimentele sunt asincrone şi sunt determinate de funcţiunile MSL, de partenerii din reţea sau de intervenţia locală a utilizatorului prin interfeţe specializate MMI;

MCL are rolul de a coordona activitatea modulelor care formează DI prin fluxul de mesaje între module, ceea ce presupune funcţiuni foarte apropiate cu cele ale unui sistem de fluxuri de lucru (*workflow*).

Un flux de lucru, workflow, constă în automatizarea procedurilor prin care documentele, informaţiile şi activităţile sunt transmise între participanţi, bazat pe un set predefinit de reguli, pentru realizarea unui obiectiv [AIT]. Workflow este liantul care leagă aplicaţiile client/server într-un continuum de procese capabile să sprijine grupurile de lucru.

Elementul central al tehnologiei managementului fluxurilor sunt pachetele de modelare de procese, care pot fi organizate în patru clase [AIT]:

- pachete "de diagramă sau desenare" care ajută în descrierea fluxurilor de lucru şi leagă descrieri textuale de procesele de lucru (*Easy-Flow, MacFlow, FlowCharting 3*);
- pachetele tradiţionale de analiză şi modelare fluxuri, care utilizează o varietate de procese predefinite şi reguli de rutare (*DEC-Linkworks*);
- pachete CASE şi software pentru inginerie industrială, care permit analiza şi modelarea pentru proiectarea şi ingineria de sistem (*SIM-Factory-CACI, Bussiness Design Facility-TI, TASC-flow-TASC, Workflow Analyser-Meta Software*);
- pachete de încredere, similare CASE, dar fiind parte a unui cadru conceptual impus utilizatorului pentru a fi aplicat.

Aceste pachete pot ajuta în descrierea proceselor individuale utilizând formalism grafic. Ele furnizează aplicaţii de statistică a proceselor şi a resurselor utilizate de acestea, ajută la cuantificarea productivităţii şi a calităţii pentru fiecare utilizator conectat, permit simularea proceselor şi a implicaţiilor tehnice generate de modificarea/revizuirea fiecărui proces. În mod ideal aceste pachete sprijină dezvoltarea unui prototip de aplicaţie, care poate fi apoi scalată în producţie.

Scopul sistemelor de fluxuri de lucru este acela de a asigura legătura dintre procese şi lumea modelării datelor [Gawlick,94].

Uzual fluxurile de lucru sunt descrise din trei puncte de vedere:

- secvenţa de activităţi;
- resursele necesare activităţilor (oameni, maşini etc.);
- fluxul de informaţii (informaţii de intrare/ieşire, control etc.).

Modelul de referinţă pentru fluxuri de lucru este WfMC, figura 4.5., iar funcţionalitatea de referinţă este obţinută prin interacţiunea dintre diferite interfeţe ale sistemului:

- pentru definirea proceselor de fluxuri de lucru;
- spre motoarele de fluxuri de lucru, ca sisteme de calcul care execută şi controlează procesele;
- spre aplicaţii apelate (existente uzual);

- spre aplicaţii de fluxuri de lucru client, care acţionează ca interfaţă pentru un agent (om sau alt calculator) spre sistemul de management al fluxurilor de lucru;
- pentru administrarea şi monitorizarea proceselor de fluxuri de lucru.

Acest model se poate utiliza ca model pentru MCL deoarece:

- este un model de referinţă cunoscut;
- WfMC conţine o componentă utilă definirii unui model de fluxuri, care poate fi utilizat de MCL ca model intern al companiei;
- există deja un limbaj formal WPDL (Workflow Process Definition Language) potrivit pentru a reprezenta modelul intern al companiei;
- motorul de workflow este un executant al modelului de workflow. Modulul local de coordonare poate fi văzut ca un motor de workflow specificat;
- modulele din DI pot fi văzute ca şi clienţi ai aplicaţiei de workflow;
- se poate configura un mediu multi-VE dacă modelul intern al companiei este configurat utilizând câte un model de workflow pentru a descrie fiecare integrare a companieie într-o VE.

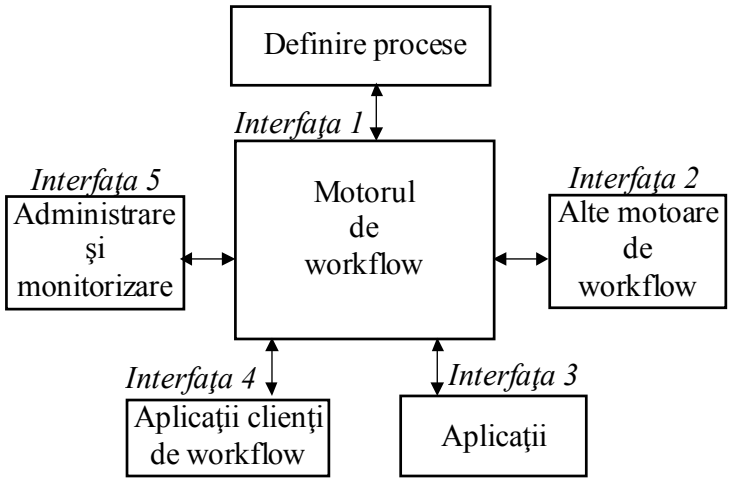


Fig. 4.5. Modelul de referinţă pentru fluxuri de lucru

Utilizând limbajul WPDL, se poate stabili modelul intern al companiei, specificând Ri – reţelele din care face parte, partenerii în fiecare reţea, lista aplicaţiilor, tipul datelor utilizate, lista activităţilor etc:

WPDL
WORKFLOW ‘ VE-R1’

NAME
DESCRIPTION
.....
//PARTICIPANT LIST
//APPLICATION LIST
//PROCESS RELEVANT DATA LIST
//ACTIVITY LIST
//TRANSITION INFORMATION LIST
END_WORKFLOW

WORKFLOW ‘ VE-R2’
NAME
DESCRIPTION
.....
//PARTICIPANT LIST
//APPLICATION LIST
//PROCESS RELEVANT DATA LIST
//ACTIVITY LIST
//TRANSITION INFORMATION LIST
END_WORKFLOW
.....
.....
WORKFLOW ‘ VE-Rn’
NAME
DESCRIPTION
.....
//PARTICIPANT LIST
//APPLICATION LIST
//PROCESS RELEVANT DATA LIST
//ACTIVITY LIST
//TRANSITION INFORMATION LIST
END_WORKFLOW

MLC este un motor particular de workflow care:

- execută modelul intern, particular al întreprinderii;
- execută/manipulează fiecare mesaj/eveniment sosit la DI;
- monitorizează tratarea fiecărui mesaj/eveniment în conformitate cu fluxul său predefinit;
- solicită decizia umană pentru mesaje necunoscute/neprevăzute;
- apelează componentele responsabile pentru manipularea mesajelor/ evenimentelor;
- administrează comportarea DI.

3. Modulul de interfaţă specifică, EDI şi STEP

- receptionează/formatează mesajele sosite/transmise în format EDIFACT. Se fac verificări de sintaxă, de integritate conţinut mesaj la recepţie şi se formatează semnalul de ieşire;
- detectează şi extrage din mesajele EDI informaţiile în alte formate, de exemplu: STEP, asociate unei comenzi;
- asigură transmisia/recepţia fişierelor în format STEP.

4. Modulul de infrastructura de comunicaţii

- asigură legătura de comunicaţii cu alte noduri din reţea;
- asigură selectarea/adaptarea parametrilor de comunicaţii în reţea;
- asigură confidenţialitatea datelor prin codare;
- asigură securitatea sistemului prin blocarea accesului neautorizat.

Problema comunicaţiilor trebuie analizată din punct de vedere al comunicaţiilor în interiorul nodului şi în exteriorul nodului. Protocolul intern de comunicaţii defineşte câteva clase de mesaje şi comenzi care asigură legătura dintre MSL şi DI, care pot fi structurate ca mesaje/informaţii despre: configurare, management, comenzi, calitate, standard (STEP, EDI), managementul sistemului informatic sau date neformatate. La nivelul unui nod, DI–ca interfaţă neutrală–comunică cu interiorul nodului pe baza unui protocol intern la care se pot alinia echipamentele specifice locale. Comunicaţia cu exteriorul se face prin „protocolul extern" de comunicaţii.

Se au în vedere două aspecte:

- comunicaţia cu alte DI, comunicaţie „standard", predefinită;
- comunicaţia cu alte „noduri" care este orientată pe protocoale standard EDIFACT şi STEP.

Infrastructura de comunicaţii care susţine conceptul de PVE trebuie să ofere ***transparenţa*** necesară utilizării unei game extinse de protocoale de comunicaţii şi trebuie să ofere facilităţi de siguranţă, adică fiecare nod să aibă implementate mecanisme specifice. Lăţimea de bandă a canalului de comunicaţii şi latenţa sunt elemente esenţiale ale unei infrastructuri de comunicaţii. În cazul interconectării laboratoarelor problema securităţii schimbului de date este importantă, pentru că asigură funcţiunile PVE.

În [Camarinha, 97] se propune o clasificare a celor mai importante aspecte legate de securitatea schimbului de date:

- accesul la informaţie fară autorizare;
- alterarea/modificarea informaţiei fară a avea autorizaţie;
- utilizarea neautorizată a resurselor;
- distrugerea datelor fără motiv (vandalism informatic).

Pentru a preveni o astfel de situaţie se va utiliza un set de obiective de securitate:

- ***controlul accesului;***
- ***autentificarea*** - siguranţa asupra declarării identităţii interlocutorului;
- ***integritate***- garanţia că se recepţionează ceea ce s-a transmis;
- garantarea ca cel ce transmite/recepţionează nu va interzice tranzacţia după ce ea a avut loc (***non-repudiere***);
- ***confidenţialitatea***- blocarea accesului la informaţia senzitivă, pentru cei neautorizaţi.

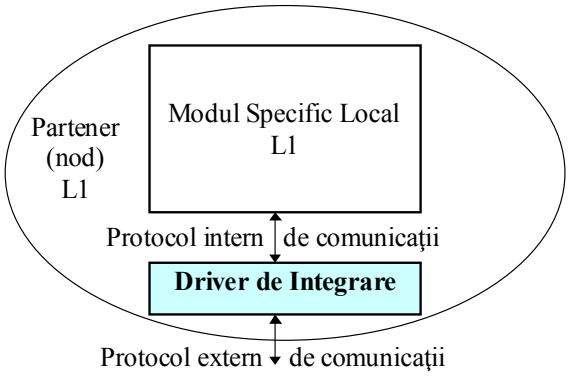


Fig. 4.6.

Aceste cerinţe conduc la utilizarea unui modul de codare/decodare şi autentificare care să ofere servicii specifice şi să asigure integritatea semnalului. Pentru PVE se au în vedere cele două tehnici importante de a asigura confidenţialitatea datelor în reţele comune de comunicaţii: codarea şi metrica-criptografia cu cheie secretă, codare asimetrică - criptografia cu cheie publică sau combinaţii între ele.

Pentru asigurarea verificării integrităţii se utilizează semnături digitale.

5. Modulul de configurare şi interfaţa utilizator

DI este configurat în funcţie de nevoile specifice ale fiecărui partener şi în funcţie de structura VE în care întreprinderea este aliat.

Modelul apelează la trei clase de informaţii:

1. Date privind compania:

- informaţii interne pentru activare/dezactivare comunicaţii cu DI;
- identificarea sistemelor interne existente, necesar a fi legate cu DI;
- date privind structura internă şi operaţiile de control (parolele utilizator şi pentru directoarele de lucru).

2. Modelul intern al companiei:

- compania operează cu mesaje în format EDI, STEP,etc.

3. Informaţii privind integrarea în VE:

- identificarea participării simultane în mai multe VE;
- informaţii privind partenerii din fiecare VE;
- identificarea partenerilor;
- adresele de reţea;
- mediul de interfaţă a partenerilor din reţea;
- drepturi de acces;
- informaţii partajate;
- configurare protocoale de comunicaţii utilizate (STEP,EDI etc.).

Laborator Virtual de Cercetare –Web Intranet/Internet

O problemă importantă legată de arhitectura PVE este aceea a soluţiilor care vor fi utilizate ca suport al schimbului de informaţii. Pentru VE, cel puţin la mometul de faţă soluţia EDI este puţin probabil să fie înlocuită cu soluţia WWW. EDI oferă mediul optim pentru manipularea unui număr mare de ordine-comenzi, în mare parte generate de sistemele proprietare PPC şi au facilităţi privind auditabilitatea (care nu sunt încă disponibile în sistemele WWW). Pentru PVE, în care prima etapă o constituie constituirea laboratorului virtual de cercetare şi didactic, soluţia orientată WWW este extrem de potrivită datorită facilităţilor multimedia oferite şi deoarece este mai adecvată tranzacţiilor iniţiate de operatorul uman.

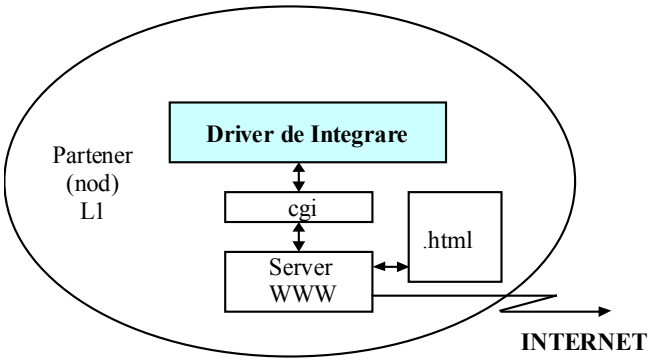


Fig. 4.7.

Adoptarea limbajelor şi a protocoalelor standard Internet se constituie în cadrul necesar funcţionării unei platforme care să permită schimbul de date şi desene 3D între parteneri aflaţi într-o alianţă bazată pe competenţe de bază complementare.

Această platformă va asigura:

- facilităţi pentru re-ingineria proceselor specifice, dezvoltarea de proiecte în comun, formarea resursei umane şi marketingul;

- reducerea duratei proiectării prin colaborarea în timp–real ca alternativă la cicluri distincte de iteraţii;
- diseminarea tehnologică şi metodologică;
- identificarea riscurilor majore şi a referinţelor într-un astfel de proces de colaborare bazată pe suport evoluat de comunicaţii, pentru aplicare trans-sectorială.

4.2. PROIECTAREA DE ALIANŢE TIP ÎNTREPRINDERE VIRTUALĂ, BAZATĂ PE MODULE FABRICATOR

4.2.1. REŢEA GLOBALĂ CUNOŞTINŢE DE REINGINERIE

Proiectarea pe bază de module a produsului are o importanţă deosebită pentru obţinerea unui răspuns rapid al companiei la cerinţele pieţii. Acest tip de proiectare pune problema utilizării de module funcţionale având furnizori diverşi, care lucrează pe platforme diferite.

Proiectarea pe bază de module, sub diferite aspecte şi abordări şi-a dovedit avantajele prin folosirea curentă în:

- ***ingineria mecanică***: programele de proiectare asistată conţin biblioteci de componente diverse, standard, care pot fi susţinute chiar de informaţii privind furnizorii;
- ***robotica***: module standard de rotaţie, translaţie din care se pot construi structuri diferite orientate client şi aplicaţie;
- ***electronica şi tehnica de calcul***: modulele electronice se constituie în elemente structurale macro-funcţionale pe baza cărora se pot construi arhitecturi diverse de sisteme de conducere şi achiziţie de date. Compatibilitatea este asigurată prin magistrale sau prin comunicaţii standardizate. Producătorii de module electronice pentru echipamente de calcul, automatizare şi comunicaţii asigură compatibilitatea acestora cu standarde existente. Producătorii de sisteme pot utiliza aceste module, optimizate ca preţ şi calitate şi pot de asemenea dezvolta alte module specilizate, care pot deveni module de uz general puse la dispoziţia altor beneficiari.
- ***reţele globale de inginerie***: in Global Engineering Network, capitalizarea cunoştinţelor porneşte de la modele pe bază cărora li se validează module reutilizabile, care înglobează capacitatea de inovaţie în reţea specifică VE.

Un mediu global de capitalizare a cunoştinţelor are ca scop sprijinirea proiectantului încă din fazele timpurii ale proiectării cu cunoştinţe inginereşti încapsulate în obiecte validate, actualizate şi rafinate într-un "lanţ logistic" continuu client-utilizator. Aceste obiecte provin din discipline inginereşti diferite (electronică, microelectronică, mecanică, software etc.). Figura 4.8. prezintă arhitectura generică a unui astfel de sistem de capitalizare a cunoştinţe [Sauer,96] .

Furnizorul deţine baza de obiecte care conţin cunoştinţele inginereşti ce sunt oferite în reţea.

Ofertantul recepţionează cererile clienţilor şi le orientează spre furnizorul cel mai potrivit. El colectează ofertele şi reţelele asociate pentru a le oferi clienţilor.

Pentru **client** tranzacţia tipică este cererea–iniţiată de utilizatorul, proiectantul–inginerul–prin care se solicită accesul la cunoştinţe dintr-un domeniu anume.

Acest sistem de reţea globală de cunoştinţe inginereşti se constituie într-un spaţiu electronic de convergenţă care asigură permanenta reactualizare a acestora şi se sprijină pe un schimb efectiv de cunoştinţe inginereşti relativ la produse, servicii, soluţii tehnice.

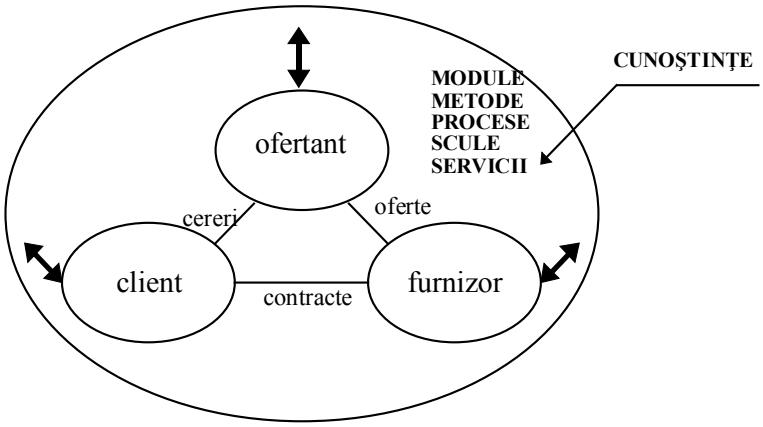


Fig. 4.8. Reţea de capitalizare a cunoştinţelor inginereşti

Dacă utilizarea modulelor sau integrarea în reţele de capitalizare a cunoştinţelor are ca scop utilizarea de entităţi generic denumite module, necesare în procesul de dezvoltare–realizare de produse/servicii, algoritmul prezentat în continuare propune tratarea, pe bază de module, a alianţelor care conduc la VE şi la capitalizarea cunoştinţelor legate de experienţa altor alianţe. Algoritmul de configurare permite selectarea diverselor configuraţii posibile de VE, capabile să asigure, în mediu simulat, optimizarea funcţiei obiectiv pentru fiecare includere a unui fabricator într-o configuraţie particulară de VE.

Din mulţimea soluţiilor posibile de VE în mediu simulat, pe baza de negociere directă între agenţii potenţiali parteneri, se poate stabili configuraţia de VE finală pentru etapa t.

Procesul este dinamic, iar satisfacerea funcţiei obiectiv pentru agenţii Fabricator este un proces complex de stabilire a unui optim ponderat între ceea ce solicită fiecare agent fabricator şi macro-obiectivele alianţei VE.

Crearea unei reţele de capitalizare a cunoştinţelor legate de potenţialii parteneri pentru alianţe virtuale se constituie într-o extensie necesară a reţelelor globale de inginerie şi generează o reţea extinsă de cunoştinţe pe care o definim ca ***reţea globală de cunoştinţe de reinginerie.***

Procesul de localizare a partenerilor focalizaţi pe un anumit tip de competenţe de bază şi având o anumită experienţă confirmată în parteneriate de afaceri încheiate cu succes, poate fi un proces care consumă timp apreciabil, care uyual nu exista atunci când se solicită un răspuns în timp real la cerinţele pieţii. Una dintre cerinţele majore ale unei pieţe în globalizare impune stabilirea unor căi, cât mai puţin consumatoare de timp, pentru a obţine informaţia necesară localizării posibililor parteneri pentru o alianţă orientată pe competenţe complementare şi încredere reciprocă. Obţinerea partenerului Fabricator sau a partenerilor constituiţi într-un lanţ logistic funcţional, validat, prin accesul la informaţiile existente într-o reţea de cunoştinţe privind posibili parteneri, poate înlocui un sistem clasic, bazat pe informaţii stocate în baze de date de contact locale. Acestea pot fi asociate cu întrega logistică privind actualizarea informatiilor reyultând o soluţie în care experienţa cumulată a multor posibili parteneri vine să „pondereze" profilul structurii Fabricator care îşi oferă serviciile.

4.2.2. SUPORT FORMAL PENTRU PROIECTAREA DE ALIANŢE TIP ÎNTREPRINDERE VIRTUALĂ

În cele ce urmează se vor folosi următoarele notaţii:

[ ]	subset
[ ]$_f$	subset fezabil f, la momentul t
{ }	set
{ }$_u$	setul care nu satisface condiţiile
Kf	constrângeri funcţionale
Cf	cerinţe funcţionale
t	iteraţia t
DDB$_F$	baza de date distribuită cu informaţii Fabricator
DDB$_{VF}$	baza de date distribuită cu configuraţii VE validate
m	numărul de configuraţii VE în DDB$_{VF}$

$\mathbf{F_{i,j}}$	Fabricatorul i din DDB_F, având proprietatea j
$\mathbf{F_{i,j}^{l}}$	Fabricatorul i, cu proprietatea j din VE^l
n	numărul de Fabricatori în întreprinderea virtuală la iteraţia t
$\mathbf{Cf_{c,p}}$	cerinţa funcţională c având proprietatea p
$\mathbf{VE^{*}}$	noua iteraţie de VE, obţinută în urma aplicării unui operator asupra unei VE
$\mathbf{VE_l^t}$	modelul de VE_l, la iteraţia t (etapa t în istoria VE_l)
$\mathbf{O_l^t}$	funcţia obiectiv la iteraţia t, pentru modelul de VE_l
α_{int}	operator de integrare partener
α_{ren}	operator de renunţare la partener
α_{rec}	operator de reconfigurare alianţa VE
α_{inl}	operator de înlocuire partener în alianţa VE
α_{conf}	operator de confirmare partener în subsetul fezabil de fabricatori
α_{inf}	operator de infirmare partener din subsetul fezabil de fabricatori

Proiectarea VE bazată pe module fabricator, dintr-un set de module fabricator candidate $\{F_{i,.}\}$, are la bază selectarea unui număr de parteneri Fabricator care să satisfacă setul de cerinţe funcţionale {Cf} , cât şi setul de constângeri impuse {Kf}pentru a maximiza funcţia obiectiv a VE

(ex. $\mathbf{O_l^t}$ **=f (cost, calitate, resurse, timp)**)

O instanţă de VE,

$$\mathbf{VE}=\bigcup_{i,j}^{n,m} Fi,. \quad \textbf{unde i=1...n; j=1...m,}$$

se constituie pe baza informaţiilor ***cantitative*** (în câte alte alianţe VE a fost inclusă, capacitatea de a oferi servicii, alinierea la standarde de calitate, etc.) cât şi informaţii ***calitative*** (rezultate ale unor colaborări în alianţe anterioare sau curente, performanţe declarate etc).

Alianţa VE este una dinamică, instanţierile succesive putând fi notate VE_l^t, unde notaţia semnifică modelul l de VE, la momentul t de generare, iar F_i^l este modulul fabricator care a fost selectat pentru alianţa l, în etapa t, iar n reprezintă numărul partenerilor din alianţa considerată.

$$\mathbf{VE}=[\bigcup_{i}^{n} Fi,.]^t = (\mathbf{F_1^l, F_2^l,...., Fn^l})^t$$

Definire operatori specifici

1. Operatorii de bază

- α_{ad} ***: adăugare agent F în alianţa VE***
- α_{ren} ***: renunţare agent F din alianţa VE:***

Se poate începe negocierea cu un agent $F_{i,j} \in \{F_{i,.}\}$ pentru integrarea în VE în urma analizei performanţelor calitative şi cantitative existente în baza de cunoştinţe distribuită $\mathbf{DDB_F}$ sau, în cazul unui nou candidat, pe baza fişei de prezentare sau a recomandărilor altor parteneri de afaceri ai acestora.

$$\alpha_{ad}\ (\ F_{m,.},\ k,\ VE\ ) : VE \rightarrow VE^{*},\ \text{unde}$$

$$VE^{*} = VE \cup \{F_{m,.}\}$$

Se poate renunţa la un agent $F_{i,j} \in \{F_{i,.}\}$ din VE în urma prestaţiei nesatisfacatoare a acestui fabricator.

$$\alpha_{ren}\ (\ F_{m,.},\ VE\ ) : VE \rightarrow VE^{*},\ \text{unde}$$

$$VE^{*} = VE \setminus \{F_{m,.}\}$$

2. Operatori derivaţi

Pentru a prezenta aspectele complexe legate de caracterul dinamic al configurării alianţei VE se consideră operatorii compuşi:

- α_{rec} ***: reconfigurare alianţa VE***
- α_{inl} ***: înlocuire agent din alianţa VE***

şi variante ale acestora, care să verifice fezabilitatea rezultatului obţinut.

Operatorul de reconfigurare VE, α_{rec}**,** aplicat la două configuraţii de întreprindere virtuală, VE_1, VE_2 permite schimbul de seturi de parteneri între cele două modele. Acest operator are avantajul scurtării duratei de configurare a VE prin utilizarea de sub-lanţuri fabricator validate deja (de exemplu: sublanţuri logistice existente şi funcţionale).

$$\alpha_{rec}\ (\ j,\ VE_1,\ VE_2\ ) : VE_1 \rightarrow VE_1^{*}$$

$$VE_1^{*} = VE_1 - \bigcup_{i=j}^{n} \alpha ren\ (\ F_{i,.},\ VE_1) + \bigcup_{i=j}^{n} \alpha ad\ (\ F_{i,.},\ VE_2,\ VE_1)\ ,$$

unde j=...semnifică ,,începând cu partenerul j din VE”

Operatorul propus nu impune constrângeri sau verificări relativ la fezabilitatea VE reconfigurată, reconfigurarea având ca suport numai informaţiile din baza de cunoştinţe referitoare la fabricatori şi alianţe VE. Pentru procesele de afaceri acest tip de operator este numai teoretic utilizabil, realitatea impunând un proces incremental prin care fiecare nou partener este integrat în VE numai după ce este verificată fezabilitatea alianţei.

Algoritmul incremental propus poate fi numit ***operator de reconfigurare condiţionată şi*** este:

La iteraţia **t**, având la dispoziţie informaţiile din bazele de cunoştinţe privind fabricatorii şi VE, $\mathbf{DDB_F^t}$, respectiv $\mathbf{DDB_{VF}^t (VE_1^t, VE_2^t,....., VE_m^t)}$ **:**

START

- **Pasul 1***: se introduc caracteristicile alianţei* VE_a^t *care trebuie reconfigurate*

- **Pasul 2***: se selectează din* $\mathbf{DDB_{VF}^t}$ **configuraţia** VE_b^t **cea mai apropiată de** VE_a^t

- **Pasul 3:** *se selectează tipul de partener* $F_{j,.}$ *care se doreşte schimbat din setul de parteneri din* VE_a^t

- **Pasul 4:** *se utilizează operatorii de bază pentru a face înlocuirea de partener*

$$\mathbf{VE_a^{t'} = VE_a^t - \alpha_{ren} (F_j^a{}_{,.}, VE_a^t) + \alpha_{ad} (F_j^b{}_{,.}, VE_b^t, VE_a^t)}$$

- **Pasul 5:** *se verifică dacă* $VE_a^{t'}$ *este fezabilă, respectiv dacă introducerea noului partener verifică condiţiile cerute de alianţa VE. Dacă este asigurată viabilitatea:*

$$\mathbf{\{F_{j,.}\} \rightarrow \{F_{j,.}\} \cup F_{j,.}^b}$$

şi

$$\mathbf{\{ VE^t\} \rightarrow \{ VE^t\} \cup VE_a^{t'}}$$

Dacă este asigurată viabilitatea se continuă , dacă nu, se renunţă la pasul curent de reconfigurare şi se incrementează contorul j.

- **Pasul 6:** *dacă* $VE_a^{t'}$ *este fezabilă se incrementează contorul j=j+1, se verifică încheierea reconfigurării,* $j<n+1$ *şi se trece la* **pasul 4,** *sau se încheie operaţiunea.*

- **Pasul 7:** $VE_a^{t'}$ *este o configuraţie nouă de întreprindere virtuală, verificată din punct de vedere al fezabilităţii. În momentul în care* $VE_a^{t'}$ *este validată, partenerii* $F_{i,j}$ *sunt validaţi urmând a face parte din setul de parteneri având atributele j.*

STOP

Operatorul de înlocuire, α_{inl}, aplicat unei VE are rolul de a anula legătura cu un fabricator $F_{i,j}$ din alianţă şi înlocuirea lui cu un alt partener, care face parte din setul de parteneri fezabili $[F_{i,.}]_f$ al mulţimii totale a partenerilor $\{F_{i,.}\}$:

$$\mathbf{\alpha_{inl}(i,VE) : VE \rightarrow VE^*}$$

$$VE^{*} = VE - \alpha_{ren}\,(F_{j,.}, VE) + \alpha_{ad}\,(F_{j,.}, [F_{i,.}]_f, VE)$$

Operatorul propus nu asigură verificarea fezabilităţii noii alianţe. Se propune un algoritm care să asigure ***înlocuirea condiţionată*** a unui partener:

La iteraţia **t**, având la dispoziţie informaţiile din bazele de cunoştinţe privind VE, $VE_1^t, VE_2^t, \ldots, VE_m^t$, mulţimea fabricatorilor $\{F_{i,.}\}$, setul fezabil de fabricatori (fabricatorii care au fost confirmaţi într-o alianţă cunoscută) $[F_{i,.}]_f$ se procedeaya astfel:

START

- **Pasul 1***: se introduc caracteristicile alianţei VE_a^t care trebuie reconfigurată şi datele unui fabricator din alianţă, care trebuie înlocuit, $F_{i,.}$*

- **Pasul 2***: se utilizează operatorii de bază pentru a face înlocuirea de partener*

$$VE_a^{t'} = VE_a^t - \alpha_{ren}\,(F_i^a{}_{,.}, VE_a^t) + \alpha_{ad}\,(F_i^a{}_{,.}, [F_{i,.}]_f, VE_a^t)$$

- **Pasul 3***: se verifică dacă $VE_a^{t'}$este fezabilă, respectiv dacă introducerea noului partener verifică condiţiile cerute de alianţa VE. Dacă este asigurată viabilitatea atunci:*

$$\{VE^t\} \rightarrow \{VE^t\} \cup VE_a^{t'}$$

şi se încheie operaţia; dacă nu, se renunţă la pasul curent de reconfigurare, şi se caută alt înlocuitor pentru $F_{i,.}$ (revenire la **pasul 2***)*
STOP

3. Alţi operatori

- α_{conf} operator de confirmare partener în subsetul fezabil de fabricatori

$$\alpha_{conf}\,(F_{i,j}) : \{F_{i,.}\} = \{F_{i,.}\} \cup F_{i,j}$$
$$\text{si}$$
$$[F_{i,j}]_f = [F_{i,j}]_f \cup F_{i,j}$$

Operatorul permite introducerea unui modul Fabricator în baza de date cu parteneri fezabili (deci automat în lista partenerilor posibili), fără verificări de fezabilitate speciale.

- α_{inf} operator de infirmare partener din subsetul fezabil de fabricatori

$$\alpha_{conf}\,(F_{i,j}) : \{F_{i,.}\} = \{F_{i,.}\} \setminus F_{i,j}$$
$$\text{si}$$
$$[F_{i,j}]_f = [F_{i,j}]_f \setminus F_{i,j}$$

Operatorul permite scoaterea unui modul Fabricator din baza de date cu parteneri fezabili (deci automat în lista partenerilor posibili), fără verificare. Infirmarea unui partener are implicaţii majore asupra bazei de date cu module VE, operatorul având opţiunea de a asigura infirmarea fabricatorului considerat din toate VE, şi înlocuirea lui cu alt partener fezabil sau lăsarea nemodificată a structurii VE deja existente, validate în baza de date, cu precizarea că în modelul VE în care s-au operat modificări ,,off-line" se marchează acest lucru, urmând ca la apelare să urmeze procesul de validare on-line a noii configuraţii. Indiferent de decizie, fabricatorul considerat este menţinut "inactiv" în baza de cunoştinţe referitoare la parteneri, fiind menţionat motivul şi argumentele dezactivării acestui agent.

Algoritm de proiectare a întreprinderii virtuale bazat pe module fabricator

START

- **Pasul 1:** ***Rafinarea scopului iniţial al VE*** pentru obţinerea descompunerii în subscopuri, până la nivel de acţiuni "elementare" posibil de alocat (VF). Descompunerea se concretizează în seturi de cerinţe funcţionale $\{Cf_{c,p}\}$, iar fiecare din aceste cerinţe $Cf_{c,p}$ trebuie să poată fi satisfăcute de module fabricator fezabile.

- **Pasul 2:** ***Selectarea subsetului de fabricatori sau alianţe VE*** care satisfac la momentul curent t, una sau mai multe cerinţe funcţionale, prin consultarea $\mathbf{DDB_F}$ **şi** $\mathbf{DDB_{VF}}$. Rezultatul acestui pas al algoritmului este stabilirea unor seturi de parteneri fezabili, $[F_{i,.}]_f$, care să satisfacă setul de cerinţe funcţionale **c**, cu atributele **p,** setul de cerinţe funcţionale stabilite prin specificaţiile de proiectare.

  - $(Cf_{c,.}) \rightarrow [F_{i,.}]_f$
  - $\{Cf_{c,.}\} \rightarrow \{[F_{i,.}]_f\}$

  Dacă se notează cu $\{Cf_{c,.}\}_u$ mulţimea cerinţelor funcţionale care nu sunt satisfăcute de Fabricatorul sau VE ales, atunci, se face iniţializarea :

$$\{Cf_{c,p}\}_u = \{Cf_{c,p}\}$$

  2.1. În urma alocării se stabileşte dacă $\{Cf_{c,.}\}_u =^? \phi$.

dacă da, înseamnă că din baza de cunoştinţe distribuită s-au putut selecta clase de parteneri capabili să satisfacă în totalitate mulţimea cerinţelor funcţionale şi se poate trece la pasul de configurare a alianţei VE(**pasul 3** al algoritmului); **dacă nu,** se selectează o cerinţă funcţională **($Cf_{c,.}$)** din mulţimea cerinţelor funcţionale şi se continuă.

2.2. Se alege un subset din setul de module Fabricator fezabile, **$[F_{i,.}]_f$** care să satisfacă cerinţele funcţionale pentru fiecare atribut, **($Cf_{c,.}$)**. În conformitate cu [Grady, 97] şi [Suh, 88] (care au studiat proiectarea bazată pe module pentru domeniul echipamentelor electronice), pentru satisfacerea celor **n** cerinţe funcţionale se pot selecta **m** dintre modulele candidat, pe baza unei asocieri **n:m**, ceea ce conduce la structuri posibile de VE în care fiecare atribut **p** din $Cf_{c,p}$ este satisfăcut de atributul **j** al Fabricatorului **i**, $F_{i,j}$,

$$\{VE\}=\{[\bigcup_{i}^{n} Fi,.]\}, \text{ unde } F_{i,.} \in [F_{i,.}]_f$$

Chiar dacă, din punct de vedere al simplităţii ar fi de preferat o asociere 1:1, Fabricatorii din setul fezabil sunt structuri care satisfac simultan o clasă de cerinţe funcţionale, şi astfel, în configurarea alianţei virtuale asocierea se face din considerente legate de maximizarea funcţiei obiectiv a VE. Formal, un modul Fabricator având **k** atribute poate satisface maxim **k** cerinţe funcţionale din $Cf_{c,p}$ impuse. Dacă s-a găsit un partener care să satisfacă cerinţa functională, se şterge cerinţa funcţională curentă din setul de cerinţe funcţionale neîndeplinite,

$$\{Cf_{c,.}\}_u = \{Cf_{c,.}\}_u \setminus (Cf_{c,.})$$

şi se reia de la punctul **2.1.**

2.3. Dacă nu se poate găsi un partener în **$[F_{i,.}]_f$,** iar răspunsurile la cererile de ofertă nu sunt satisfăcătoare, setul de cerinţe funcţionale nu poate fi satisfăcut. Algoritmul se poate încheia prin decizia că nu se poate stabili o configuraţie VE, care să satisfacă cerinţele funcţionale impuse sau poate continua prin relaxarea cerinţelor funcţionale sau prin reformularea lor. În cazul participării la un BID imposibilitatea satisfacerii tuturor cerinţelor funcţionale conduce la lipsa eligibilităţii alianţei VE, caz în care algoritmul se încheie cu menţionarea cauzelor.

2.4. Reconfigurare cerinţe funcţionale şi reia algoritmul de la punctul **2.1**.

- **Pasul 3:** ***Stabilirea de varinte VE*** utilizând cele **m** seturi de module fezabile care formează mulţimea de seturi fezabile obţinută la pasul anterior. Pasul

curent are rolul de a genera **v** variante de structuri VE alternative, care să satisfacă setul de cerinţe funcţionale relevante impus.
Se face notaţia:

$$\{[F_{i,.}]_f\}_u=\{[F_{i,.}]_f\}$$

3.1. Se testează dacă au fost prelucrate toate seturile $[F_{i,.}]_f$ disponibile

$$\{[F_{i,.}]_f\}_u=\varphi.$$

dacă da, se trace la ***pasul 4*** al programului. Se selectează urmatorul set fezabil de module $[F_{i,.}]_f$, şi se scoate din $\{[F_{i,.}]_f\}_u$.

3.2. Se prelucrează setul curent $[F_{i,.}]_f$ în vederea generării structurii de VE. Se testează dacă toate modulele Fabricator din setul curent au fost integrate în structura de Fabricator în dezvoltare:

$$[F_{i,.}]_f=\varphi$$

dacă da, modelul de VE_l^t curent este realizat şi se trece la ***pasul 4***

3.3. Se selectează un modul $F_{i,.}$ şi

3.3.1. Aplică operatorul α_{ad} ***cu verificarea condiţiei de viabilitate a VE obţinut***. Dacă structura este viabilă se repeta **3.2.,** dacă nu, setul curent $[F_{i,.}]_f$ este marcat pentru ştergere ca set nefezabil pentru secvenţa curentă de generare VE şi se trece la punctul **3.1.**

3.4. Se minimizează VE_l^t verificând dacă în structura VE diverse cerinţe funcţionale sunt satisfăcute de acelasi partener (scăderea gradului VE generat).

3.5. Verifică dacă modelul generat verifică setul de constrângeri impus. Dacă cel puţin o constrângere este neîndeplinită selectează alt set din $\{[F_{i,.}]_f\}_u$ **şi** alege modulul $F_{i,.}$ care satisface cerinţa funcţională cu probleme. Se sare la punctul **3.3.1.** Dacă nici o mulţime fezabilă Fabricator nu poate oferi soluţie, varianta curentă de VE nu este introdusă în setul fezabil generat la momentul t, $[VE^t]_f$, şi se sare la punctul **3.1.** Altfel se sare la punctul **3.1.**

- **Pasul 4:** ***Evaluarea variantelor obţinute şi stabilirea de noi variante prin alicarea operatorilor compuşi.***

4.1. Se stabileşte numărul de iteraţii dorite.

4.2 S-a realizat numărul de iteraţii dorit (stabilit de operator)**:**
dacă da, se ordonantează structurile de VE din {[VE^t]} în funţie de rezultatul evaluării şi se trece la punctul **5**;

4.3. Evaluează $[VE^t]_f$ prin prisma funcţiei obiectiv :

$$\mathbf{O_1^t = f\ (cost,\ calitate,\ resurse,\ timp)}$$

4.4. Aplică operatorii compuşi, α_{rec} **şi** α_{inl} cu verificarea fezabilităţii, pentru generare de structuri VE alternative, $[VE^{t'}]_f$. Se notifică iteraţia curentă şi se revine la punctul **4.2.**

- **Pasul 5 :** ***Definitivarea alianţei.***

5.1. Se notifică partenerilor includerea într-o alianţă şi se cere confirmarea de participare, în ordinea în care VE au satisfăcut funcţia obiectiv.

5.2. Se aşteaptă răspunsul partenerilor şi se stabileşte alianţa câştigătoare.

5.3. Se negociază cu partenerii aspectele de detaliu.

5.4. Se definitivează alocarea de sarcini.

Stop

Testarea şi validarea algoritmului s-a făcut utilizând o variantă modificată a programului ***KnowledgeNet*** al companiei americane IMS (Internet Media Service), program dezvoltat la cererea firmei HP, pentru a putea localiza, în timp real, specialişti având cunoştinţele şi expertiza necesară participării în diferite proiecte sau pentru a putea fi implicaţi ad-hoc în echipe de cercetare multi– şi interdisciplinare.

KnowledgeNet este bazat pe facilităţile Web şi poate căuta după diferite criterii entitatea solicitată, asigurând suportul necesar organizării echipelor de lucru, iar în cazul specialiştilor aflaţi în locaţii diferite geografic, constituirea echipelor virtuale de lucru. Fiind o soluţie orientată browser, ea este accesibilă indiferent de amplasarea geografică a celor ce utilizează programul şi indiferent de sistemul pe care îl utilizează, dacă acesta oferă facilităţile de interconectare prin intermediul unei reţele intranet sau internet. Aplicaţia este stand-alone, de tip groupeware. În Anexa1 este prezentată interfaţa cu utilizatorul, structura utilizată pentru prezentarea profile-ului unui modul Fabricator, domeniile după care browser-ul asigură căutarea.

4.2.3. BIBLIOTECĂ CU ACCES INTELIGENT LA CUNOŞTINŢE DE REINGINERIE

Problema creării si organizării bibliotecilor cu acces inteligent, ca reţele globale de cunoştinţe inginereşti şi prin integrarea lor în reţele globale de cunoştinţe pentru reinginerie, are ca obiectiv stabilirea unei ***infrastructuri semantice comune*** care să permită întreprinderilor să combine eficient cunoştinţele interne cu cele din exterior, şi să utilizeze experienţa validată a alianţelor virtuale stabilite în timp, prin intermediul unei reţele globale de inginerie şi informaţii despre alianţele virtuale validate, care să asigure:

- ***cadrul semantic pentru sistematizarea cunoştinţelor inginereşti***, care în general nu sunt prezente în forme care să asigure utilizarea în reţea;
- ***cadrul semantic pentru sistematizarea cunoştinţelor de re-inginerie*** a afacerilor şi creare de alianţe virtuale;
- ***setul de pachete software*** pentru achiziţie, prelucrare şi prezentare a cunoştinţelor inginereşti şi a informaţiilor legate de ,,comportarea" partenerilor în VE;
- ***infrastructură necesară capturării, partajării şi întreţinerii cunoştinţelor specifice întreprinderii***, de la proiectare şi dezvoltare produse şi procese la re-ingineria afacerilor, a sistemului, ca parte a sistemelor complexe bazate pe alianţe virtuale.

Argumentele pentru o astfel de reţea de cunoştinţe de reinginerie sunt:

- ***proiectarea optimă*** necesită informaţii obţinute la anumite momente de timp;
- ***reutilizarea cunoştinţelor de proiectare şi a serviciilor*** conduce la creşterea productivităţii şi la micşorarea duratelor de proiectare, cu efecte majore legate de durata apariţiei pe piaţă ,,time to market" a unui produs nou;

În domeniul proiectării se consideră oportună reutilizarea şi reconfigurarea proiectelor existente, obţinute din propria bază de cunoştinţe (experienţă) sau preluate din exterior (licenţe). Adesea însă proiectele disponibile nu sunt într-un format potrivit şi trebuie reproiectate. Existenţa unui format comun ar micşora timpii necesari conversiilor şi ar creşte disponibilitatea de timp pentru proiectarea creativă.

- ***integrarea firească a informaţiilor*** interne şi externe facilitează echipa de proiectare.

Proiectanţii lucrează în echipe distribuite geografic, agenţi autonomi, între care există un flux constant de informaţii pentru negociere şi schimb de rezultate. Camera de lucru electronică, laboratorul virtual de proiectare, respectiv echipele virtuale de proiectare precum şi stocarea documentelor reprezintă un suport

important pentru managementul de proiect. Paradigma reutilizării cunoştinţelor pentru construirea unui sistem multiagent (figura 4.8.) subliniază importanţa identificării şi apoi utilizării unor cunoştinţe generale prin înglobarea lor într-un mediu de dezvoltare, care să sprijine proiectantul oferindu-i cunoştinţe precodate în locul soluţiei de a trebui să-şi reprezinte, într-un mod propriu totalitatea cunoştinţelor pe care le utilizează [Jennings, 94].

- ***suport pentru întreg ciclul de viaţă al produsului***, cercetare/dezvoltare/ proiectare/producţie/testare/support post-vânzare/întreţinere; accesul la informaţiile specifice fiecărei faze din ciclul de viaţă al produsului asigură suportul pentru inovaţie şi se constituie într-un mijloc care sprijină creşterea calităţii produselor viitoare;
- ***necesităţile industriei***: proiectarea inginerească este într-un proces de continuă schimbare în efortul de a creşte eficienţa, flexibilitatea şi de a optimiza timpul de răspuns la cerinţele pieţii;
- ***reutilizarea cunoştinţelor referitoare la comportarea partenerilor*** în alianţe virtuale şi permanenta reactualizare a acestor cunoştinţe se constituie într-un suport care susţine răspunsul rapid, cu incertitudini mimime, la cerinţele pieţii.

Pentru asigurarea acestor cerinţe Fabricatorii trebuie:

- ***să crească nivelul de utilizare/reutilizare al cunoştinţelor interne şi externe;***
- ***să se concentreze pe competenţele de bază*** şi să ţină cont de serviciile disponibile pe piaţă în decizia de a produce-cumpara (***make/buy***), pe baza analizei cost/performanţă;
- ***să participe la alianţe virtuale***, prin integrarea în lanţuri logistice, care sunt focalizate pe competenţe de bază complementare.

O infrastructură semantică comună pentru un astfel de sistem conţine:

- ***un set comun de "scule"*** care să poată fi utilizate la nivelul fiecărui agent-partener cât şi între agenţi;
- ***un cadru logic pentru sistematizarea cunoştinţelor*** referitoare la procesele de afaceri în care agenţii Fabricatori sunt implicaţi, la nivel semantic;
- ***o arhitectură care să utilizeze aceste facilităţi*** pentru a asigura infrastructura necesară unui Fabricator pentru capturarea, partajarea, utilizarea şi întreţinerea cunoştinţelor specifice.

Pe baza experienţei din reţelele de cunoştinţe inginereşti [Buckley, 96] se propune o arhitectură pentru reţeaua de cunoştinţe de reinginerie, figura 4.10.

Aplicaţia de end-user are rolul de a formula şi transforma cererile utilizatorului în specificaţii pentru căutare de cunoştinţe inginereşti sau partener/sub-lanţuri logistice. Aceeaşi aplicaţie prezintă rezultatele căutării către beneficiar, în

conformitate cu setările făcute de acesta. Rezultatul ales, instanţă a unei soluţii posibile, va constitui un modul al modelului de aplicaţie tip alianţa virtuală sau produs/proces proiectat (atunci când cunoştinţele sunt inginereşti).

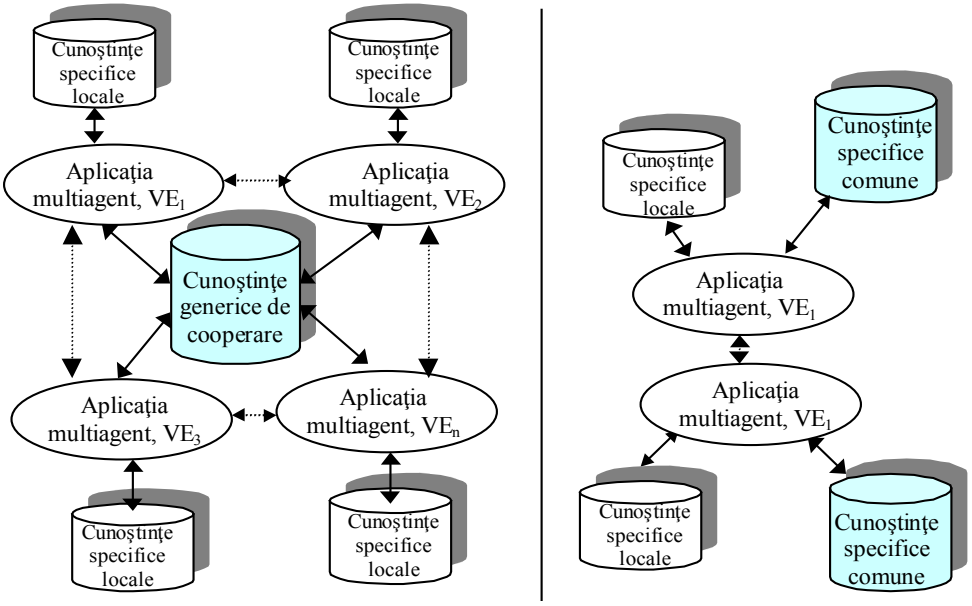


Fig. 4.9. Paradigma reutilizarii cunoştinţelor

Nucleul central al unei astfel de aplicaţii este ***motorul de căutare*** care are sarcina localizării surselor posibile de cunoştinţe şi transformarea unei specificaţii de căutare în unul sau mai multe şiruri de cereri către „repository-ii" de cunoştinţe, prin care se realizează accesul distribuit la cunoştinţele intrene şi externe.

Dicţionar de date; *articolele de date sunt identificate ca blocuri constitutive ale sistemelor de Fabricaţie având suport informatic. Companiile utilizează o mare diversitate de date ceea ce face extrem de importantă specificarea formei, tipului şi semnificaţiei acestora prin compilarea unui dicţionar de date.*

Acest dicţionar este asemănător cu toate celelalte dicţionare, conţinând definiţii (în această situaţie a datelor utilizate). Semnificaţia datelor trebuie, de asemenea, să fie foarte bine realizata, deoarece adesea termeni aparent clar precizaţi a priori capată interpretări diferite, în diferite situaţii şi pentru diferite companii. Evident în aceaşi situatie se află chiar şi termenii tehnici care adesea îşi schimbă semnificaţia odată cu schimbările din tehnologie. Un dicţionar de date conţine definiţii ale acestora. Un articol dată este in relaţii cu alte articole dată, iar acest lucru se caracterizează

prin diagrame de relaţii. Topicile sunt analizate atunci când se descriu modele asociate cu structuri de baze de date.

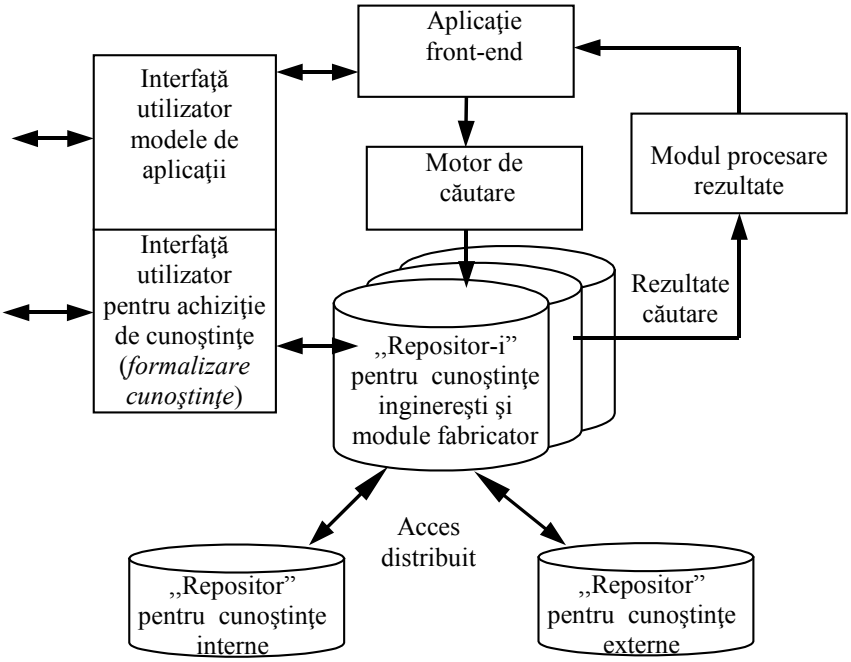


Fig. 4.10. Arhitectura sistemului de reinginerie

Repositor şi memorare de date; *datele reprezentate printr-un model trebuie să fie memorate şi accesibile utilizatorilor direcţi sau aplicaţiilor care necesită aceste date. Repositor-ul de date se conceptualizează ca interfaţă între aplicaţiile utilizator şi datele memorate într-o arhitectură de sistem informatic al întreprinderii integrate cu calculator.*

Articolele de date sunt depozitate în depozite de date. În acest stadiu depozitul de date se poate considera ca un termen conceptual, neavând încă o formă fizică sau software, cum ar fi o bază de date. Aplicaţii diverse necesită acces la date. Există date proprietare unei/unor aplicaţii, care nu necesită partajarea la nivel de companie şi pot fi depozitate în baze de date proprii aplicaţiilor. Repositorul de date are rolul de a oferi/a pune la dispoziţie utilizatorilor şi aplicaţiilor software un singur punct prin care pot accesa datele partajate ale companiei, formând o interfaţă universală. El poate fi descris ca o structură care conţine date despre date, conţinând un director al datelor partajate identificate prin definiţii ale datelor incluse într-un dicţionar. Repositor-ul trebuie, de asemenea, să cunoască locul unde sunt memorate datele, relaţiile dintre elementele de date, aplicaţiile lor.

Principala funcţiune a repositor-ului este de a izola aplicaţiile de schimbările care pot avea loc în metodele de integrare, reducând nevoia de întreţinere a aplicaţiilor.

4.3. PROBLEMA CONDUCERII SISTEMELOR DE FABRICAŢIE INTEGRATE ÎN STRUCTURI DE ÎNTREPRINDERE VIRTUALĂ

Fabricatorii, agenţi autonomi, intră în alianţe virtuale VE pe baza înţelegerii comune a afacerii, cu scopul de a-şi întări poziţia pe piaţă şi a-şi îmbunătăţi şansele de a satisface cerinţele clienţilor, aflate în schimbare rapidă. Baza pentru un parteneriat VE este încrederea reciprocă [Goldman, 94] şi existenţa unui raport tip câştigător-câştigător între parteneri.

Se pot consideră patru faze privind existenţa VE, conform [Mertens, 95]:

1. ***faza iniţială***; evaluarea afacerii şi selectarea partenerilor;
2. ***faza de configurare***: se referă la „semnare" înţelegeri, reţelarea logistică şi integrarea partenerilor;
3. ***faza de operare;***
4. ***faza de dezmembrare a alianţei virtuale, sau reconfigurare a alianţei*** după ce obiectivul de afacere a fost îndeplinit sau, dacă VE şi-a pierdut oportunitatea.

O problemă majoră la nivel de VE este aceea că funcţionarea PP&C-ului clasic, la nivelul fiecărui partener, neadaptată cerinţelor de mediu distribuit, face ca planurile la nivelul acestora să fie necontrolabile din perspectiva structurii virtuale, ceea ce conduce la suprastocuri, timpi lungi de transfer şi reducerea vitezei de răspuns al meta-sistemului la evenimente neprevăzute. Comunicaţiile în mediu electronic asigură condiţiile necesare fazelor 1, 2 şi 4. Pentru faza 3, cea de operare, este necesară crearea unui sistem de management al comenzilor (OPM-Order Production Management) care să asigure funcţionarea sistemului de-a lungul întregului lanţ de procese [Westkamper, 97].

Problema conducerii unei VE este o problemă complexă, de elaborare a unor „pachete de decizii" privind operarea ansamblului virtual bazat pe structuri Fabricatori independente astfel încât să satisfacă comenzile clienţilor în condiţii de eficienţă cost/calitate/timp.

Procesul de coordonare a comenzilor determină rutele optime pentru satisfacerea cerinţelor clienţilor. Rutele optime sunt selectate în funcţie de starea curentă a partenerilor, de operaţiile fezabile ce pot fi executate de fiecare partener la un moment dat, de capacităţile de producţie disponibile, de duratele de prelucrare specifice, de structura de cost, distribuţia geografică şi costuri logistice, de structura inventarului şi a semifabricatelor disponibile în reţea. Pentru generarea

rutei optime (planul de procese la nivelul VE) este esenţial să se stabilească agenţii Fabricatori care vor participa la execuţia comenzilor, alocarea activităţilor şi subactivităţilor specifice, care este universul de timp în care se estimează că pot face acest lucru.

Dintre aspectele importante referitoare la specificul alianţelor, necesare a fi luate în calcul sunt:

- ***independenţa partenerilor***: multitudinea de interese legate de apartenenţa posibilă la mai multe alianţe poate conduce la conflicte;
- ***poziţia egală şi autonomia partenerilor*** face imposibilă „forţarea" de comenzi sau de condiţii (mai ales timp) de onorare a unei comenzi;
- ***partenerii nu oferă întotdeauna date suficiente***, necesare procesului de coordonare la nivel de alianţă, deoarece o parte a acestora reprezintă date strategice, vitale exstenţei la nivel de structură autonomă;
- ***duratele de timp disponibile pentru negociere cu clienţi sunt limitate*** de cerinţele de răspuns în timp real, de la câteva minute la decizii imediate [Richards,97].

Optimizarea procesului de planificare a activităţilor necesare realizării unui ansamblu de comenzi necesită un timp apreciabil, uzual indisponibil atunci când apar evenimente aleatoare inevitabile la nivelul individual al partenerilor sau la nivelul altor părţi ale lanţului logistic VE. În figura 4.10. este prezentat universul

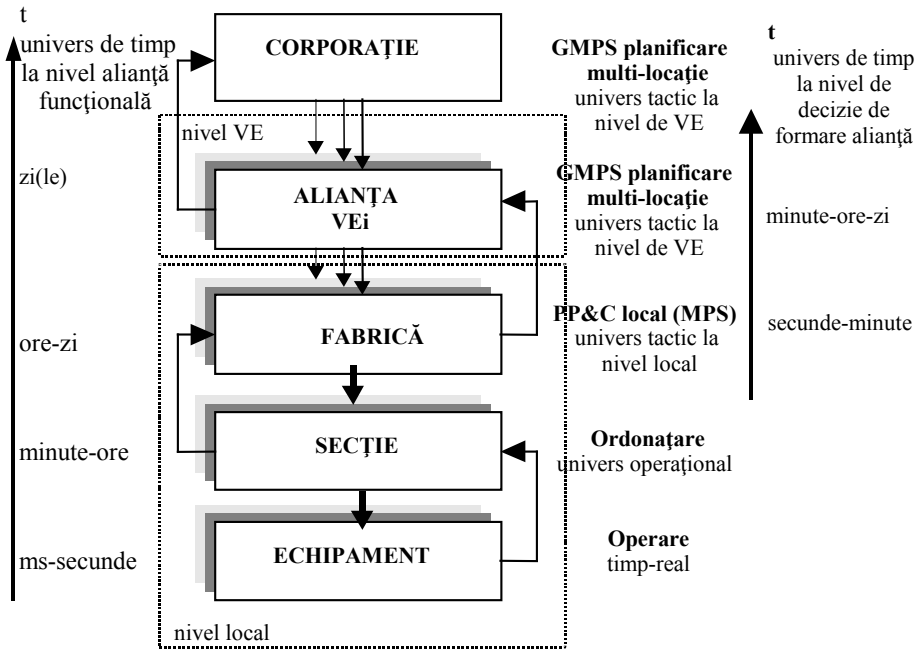


Fig. 4.10. Planificare ierarhică şi univers de timp în VE

de timp la diferite niveluri, de la cel al VE, la cel a controlului local şi până la nivel de echipament.

Este interesantă comparaţia cu „universul de timp" propus în [Albus,93] pentru nivelele unei arhitecturi ierarhice de control, la nivel de sistem de fabricaţie local şi universul de timp pentru operarea VE. În acest sens se poate observa că operarea unei VE impune constrângeri de timp real uneori similare cu cele pentru nivelurile cele mai de jos ale arhitecturii ierarhice de control, minute pentru negocierea unui contract şi ore, pâna la nivel de zi, pentru deciziile legate de ordonanţarea şi re-ordonanţarea în cazul apariţiei unor evenimente neprevăzute la unul din partenerii din alianţă.

Planificarea operaţiilor legate de derularea comenzilor în cadrul VE necesită generarea unui plan global master de ordonanţare, ***Global Master Production Schedules (GMPS),*** denumit în literatură planificarea agregată a producţiei, ***Agregate Production Planning (APP),*** care să fie preluat, la nivelul fiecărui Fabricator din alianţă, de către modulul propriu de ordonanţare principală a producţiei (***Master Production Schedules–MPS***) cu funcţiunile prezentate în cap.1. APP este situat la nivelul tactic de planificare [Browne,96], corespunzător figurii 4.11. constituindu-se în nivelul cel mai de sus al sistemului ierarhic de planificare a producţiei, ***Hierarchical Production Planning (HPP).***

Indiferent dacă APP este implementat la nivelul unui agent de coordonare şi realizează GMPS utilizând algoritmi genetici sau este distribuit în cadrul componentei proactive a modulului de planificare şi control al VE, rolul ei este de a optimiza planul de ordonanţare raportat la timp/cost/calitate şi ţinând cont de constrângerile la nivelul alianţei VE.

Pentru a optimiza procesul de planificare care necesită o reacţie rapidă, în timp real, la multitudinea de evenimente aleatoare care se produc atât la nivelul Fabricatorilor ca unităţi autonome, cât şi la nivelul VE, se propune o arhitectură (figura 4.11.) în care taskul de control este distribuit între două componente:

- ***Planificare proactivă;***
- ***Control reactiv.***

Arhitectura propusă este o consecinţă a dificultăţilor sesizate în funcţionalitatea modulelor clasice de PP&C, focalizate pe cele două direcţii specifice:

- ***Planificare***; aplicaţiile de MPS nu sunt fezabile pentru aplicaţiile în VE;
- ***Controlul***; controlul cu reacţie, specific sistemelor de fabricaţie, este clasic şi nu ţine cont de efectele induse de comportarea altor noduri din VE.

Pentru predicţie sunt utilizate informaţiile referitoare la starea curentă a comenzilor aflate în lucru şi sunt luate în calcul toate etapele, de la transmiterea ordinelor către parteneri şi până la depozitarea produselor finite.

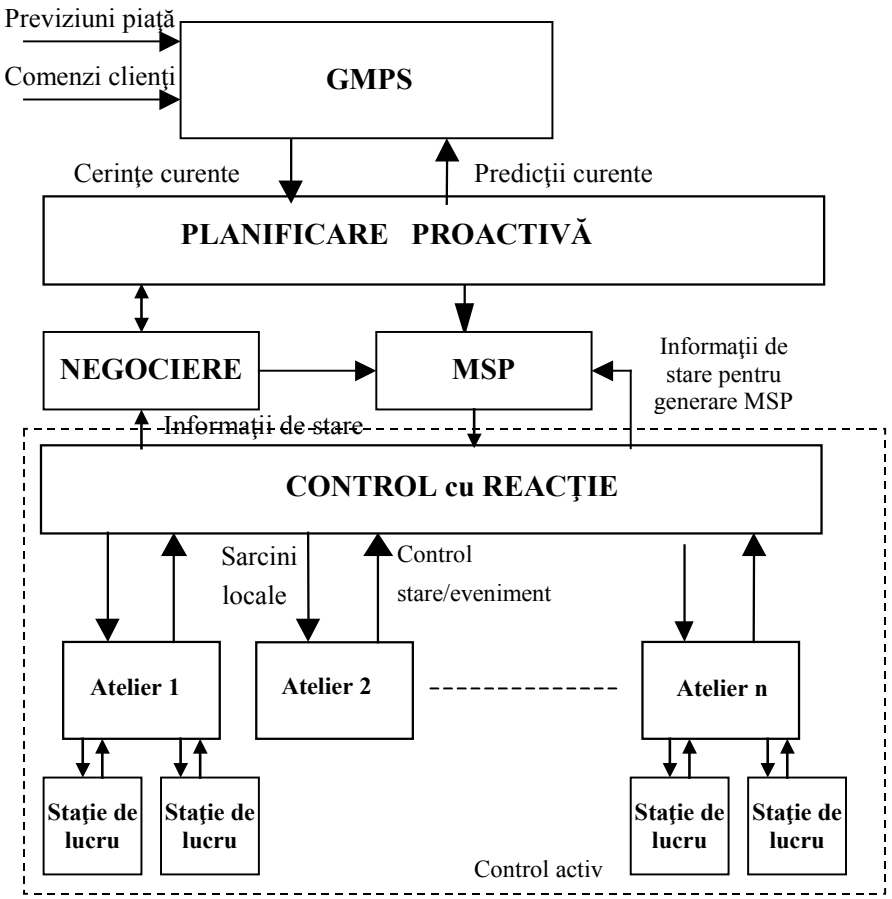


Fig. 4.11. Arhitectura sistemului de planificare şi control al alianţei VE

Controlul reactiv al fluxurilor de comenzi asigură controlul execuţiei comenzilor la nivelul fiecărui partener şi corecţiile necesare, bazate pe starea curentă şi pe evenimentele locale ale sistemului de fabricaţie.

Componenta locală proactivă are ca sarcini negocierea participării agentului la realizarea activitaţilor legate de rezolvarea comenzilor pe care alianţa le-a negociat.

Componenta reactivă are ca sacină controlul acţiunilor la nivelul operaţional: secţie, atelier. Pe baza componentelor pro-activă şi reactivă se pot realiza atât configuraţii descentralizate (figura 4.12.), specifice alianţelor de tip întreprindere virtuală cât şi configuraţii centralizate specifice companiilor multi-locaţie (figura 4.13). Utilizând tehnici de inteligenţă artificială, s-au dezvoltat sisteme bazate pe agenţi inteligenţi care comunică şi negociază în vederea rezolvării problemei de planificare şi control al ordinelor de producţie.

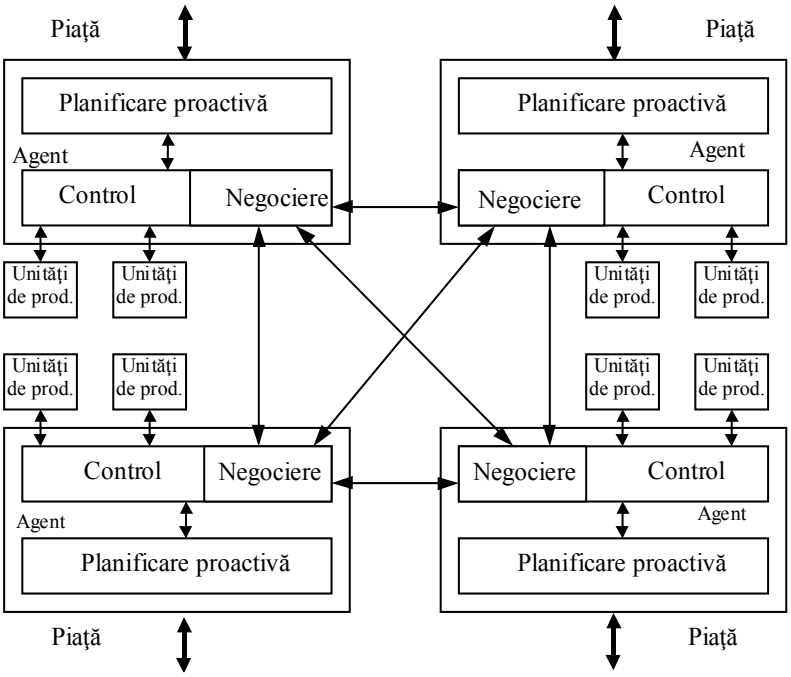


Fig. 4.12. Sistem bazat pe agenţi autonomi

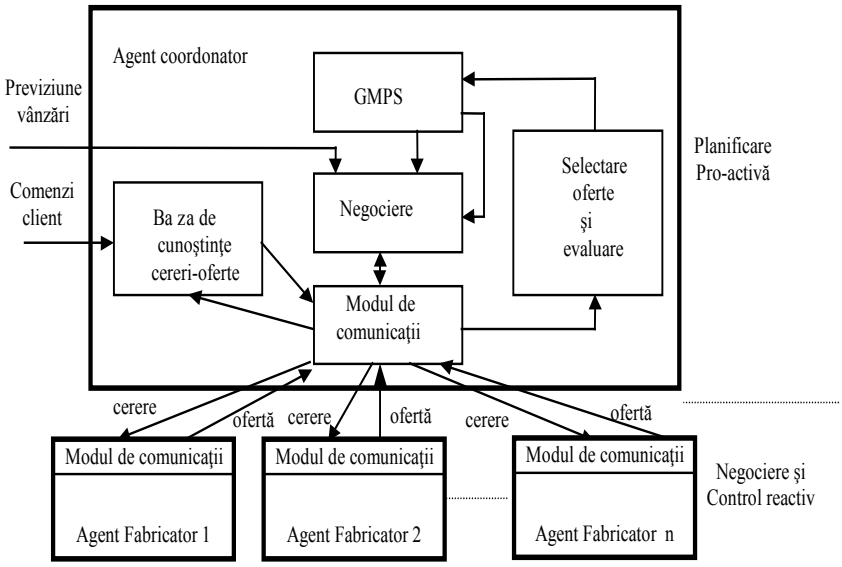


Fig. 4.13. Configuraţie bazată pe agenţi coordonaţi central

4.4. ARHITECTURĂ DE CONTROL SUPERVIZOR ŞI ACHIZIŢIE DE DATE PENTRU FABRICATOR

Pentru sinteza controlului avansat pentru ***fabricator*** este necesară satisfacerea următoarelor cerinţe:

- ***robusteţe**:* presupune conservarea proprietăţilor structurale ale DEVS (Discret Event Variable Systems), viabilitate, mărginire, reversibilitate, în prezenţa perturbaţiilor interne, defectări de echipamente, întârzieri,etc. Suportul teoretic este asigurat de teoria calitativă a supervizorului;
- ***adaptivitate (****extern-adaptiv*): robusteţea este completată cu abilitatea de adaptare la schimbările din mediul extern. Modelele orientate piaţă asigură satisfacerea acestei cerinţe [Lin., 93];
- ***autonomia sistemului de suport decizie***: creşterea rolului sistemelor co-operative, multi-agent în cadrul sinergetic al întreprinderii avansate [VanBrussel, 95], [Warnecke, 93], [Kidd, 94];
- ***completitudiea reprezentării sistemului***, ca suport pentru abordarea multiaspect a întreprinderii; sistem dinamic complex constând într-un ansamblu de resurse eterogene şi mozaic de activităţi [Camarinha, 95];
- ***conectivitatea sistemului de multimodelare:*** crearea facilităţii de a putea integra modele necesare activităţii complexe de proiectare integrată;
- ***reutilizarea modulelor funcţionale***: dezvoltarea de module software reutilizabile, adecvate să răspundă nevoilor frecvente de schimbare din sistemele de fabricaţie;
- ***conformitate software***: conformitate cu standardele şi specificaţiile de sistem deschis.

Sistemele de conducere pentru fabricator pot fi sisteme ierarhice de planificare şi control (din prisma experientei CIM) sau heterarhice (susţinute de sistemul holonic, fractal sau bionic de fabricaţie). Orizontul de timp asociat nivelurilor unei arhitecturi de conducere pentru un fabricator (planificare strategică, planificare operaţională, controlul fabricaţiei şi controlul la nivel de maşină) datorită integrării fabricatorului în întreprinderi virtuale capătă o altă semnificaţie.

Atât timp cât decizia la nivel de alianţă virtuală este o decizie care se ia în timp real, condiţiile pentru deciziile operaţionale la nivelul fabricatorului trebuie să îndeplinească cel puţin aceleaşi condiţii de timp real (chiar dacă din punctul de vedere al conducerii buclă de reglare la nivel de fabricator, fiind internă celei de reglare a funcţionării întreprinderii virtuale trebuie să aibă viteza mult mai mare). În aceste condiţii, din punct de vedere al orizontului de timp piramida decizională

[Rembold,93] devine mult mai plată, iar nivelurile de timp asociate unui sistem de control în timp real (RCS-Real-time Control System [Albus,91]) îşi modifică valorile pentru nivelurile superioare ale ierarhiei pentru a putea asigura compatibilitatea cu cerinţele de operare impuse de întreprinderea virtuală. Conducerea în timp real a fabricatorului sau operarea pe termen scurt reprezintă totuşi categoria cea mai critică de probleme, din punctul de vedere al constrângerilor de timp real.

În capitolul 1 s-a propus o arhitectură generică de control al activităţilor de producţie, iar specificarea principalelor blocuri funcţionale (monitorizare, diagnoză, ordonantare şi distribuţie) este necesară pentru crearea unui sistem de control de tip control supervizor şi achiziţie de date (SCADA). În [Caramihai,97] sunt prezentate aspectele legate de specificul conducerii în timp real a sistemelor flexibile de fabricaţie, sunt descrise funcţiile de conducere generice şi este prezentat fluxul de informaţii la nivelul unui modul generic de conducere.

Pentru fabricator se pot defini două regimuri de funcţionare asociate nivelului operaţional local:

- ***funcţionarea în regim normal***; sistemul este condus pe baza unui model de timp real (Real-Time Model, RTM), iar starea este monitorizată pe baza datelor achiziţionate din sistem şi prelucrate în vederea validării lor;
- ***funcţionarea în regim perturbat***; se pune problema modificării politicii de conducere, în special datorită modificării comportării sistemului condus.

În condiţii normale de funcţionare a sistemului apar diferenţe din ce în ce mai mari între funcţionarea sistemului real şi modelul de timp real (obţinut în urma unui proces de simulare), pe baza căruia sistemul este condus, mai ales datorită influenţei stochatice manifestate prin intermediul comenzilor de prelucrare, defecţiuniulor din sistem, variaţii ale timpului de lucru şi, evident, datorită impreciziei modelelor. Acest lucru impune o permanentă comparare a datelor reale cu cele din simularea în timp real, astfel încât modelul să poată fi modificat, ca în figura 4.14.

Bucla A este bucla de control la nivelul fabricii fizice. Bucla de control B conţine modelul fabricatorului, iar bucla de control C are drept scop controlul modelului de timp real. Fluxul de informaţii care să susţină acest sistem de control supervizor şi achiziţie de date este prezentat în figura 4.15.

Atât timp cât eroarea dintre starea curentă şi starea obţinută în urma simulării este în limitele unei valori acceptate sistemul este condus după aceeaşi lege. În momentul în care eroarea depăşeşte valoarea max. acceptată, pe baza datelor curente şi a istoriei sistemului, modelul de timp real este modificat pentru a fi „sincronizat” cu comportarea fabricatorului.

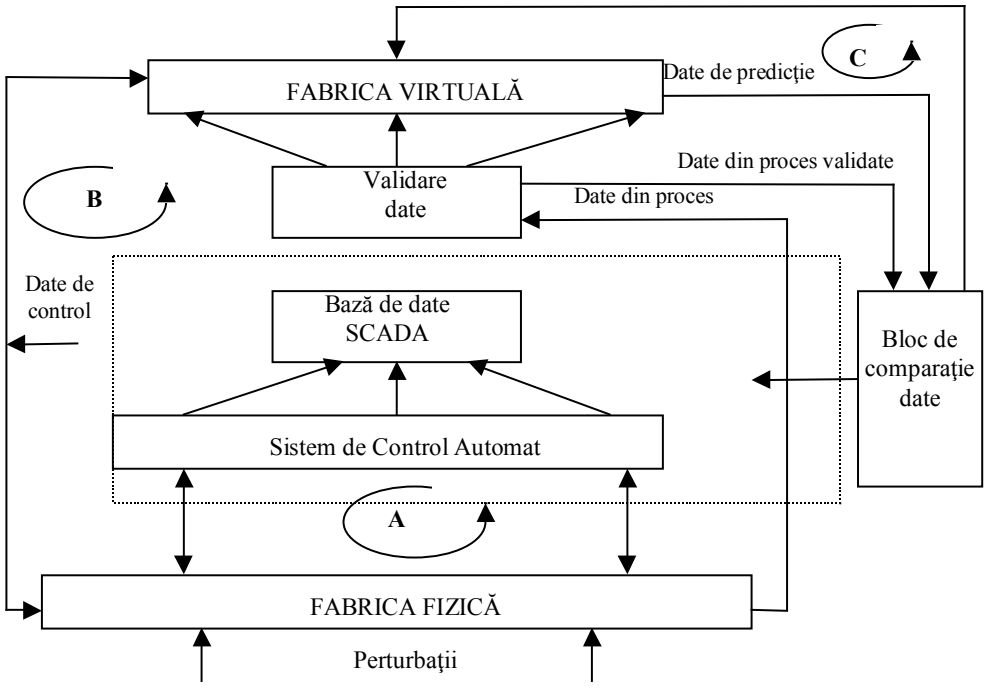


Fig. 4.14. Arhitectură de control supervizor pentru fabricator

Etapele descrise în figura 4.15 sunt corespunzătoare sarcinilor pe care trebuie să le îndeplinească un sistem de control supervizor:

- să diagnosticheze perturbaţiile;
- să genereze procedura de modificare a RTM în urma depăşirii deviaţiilor max. acceptate;
- să valideze noul model prin simulare (uzual off-line);
- să asigure schimbarea legii de control.

În [Camarinha,95] se abordează problema plasării supervizorului în contextul conducerii sistemelor de fabricaţie organizate CIM. În taxonometria propusă, între activităţile de inginerie propuse, alături de planificarea sistemului, proiectarea produselor, planificarea procese, planificarea producţie, planificarea execuţiei se află şi supervizarea fabricaţiei.

Corespunzător clasificării făcute, pentru fabricator, având un sistem ierarhic de control, se propune următoarea detaliere a activităţilor de supervizare:

- supervizor la nivel fabricator;

- supervizor la nivel de atelier;
- supervizor la nivel de celulă de fabricaţie;
- supervizor la nivel de echipament.

Pentru o arhitectură de control heterarhică (de exemplu într-un sistem holonic) controlul supervizor se realizează la nivelul holonului sau a holarhiei.

Pe baza deviaţiilor observate în sistem, controlul supervizor poate anticipa situaţii viitoare de eroare, utilizându-se în acest sens tehnici de diagnoză pe baza modelului.

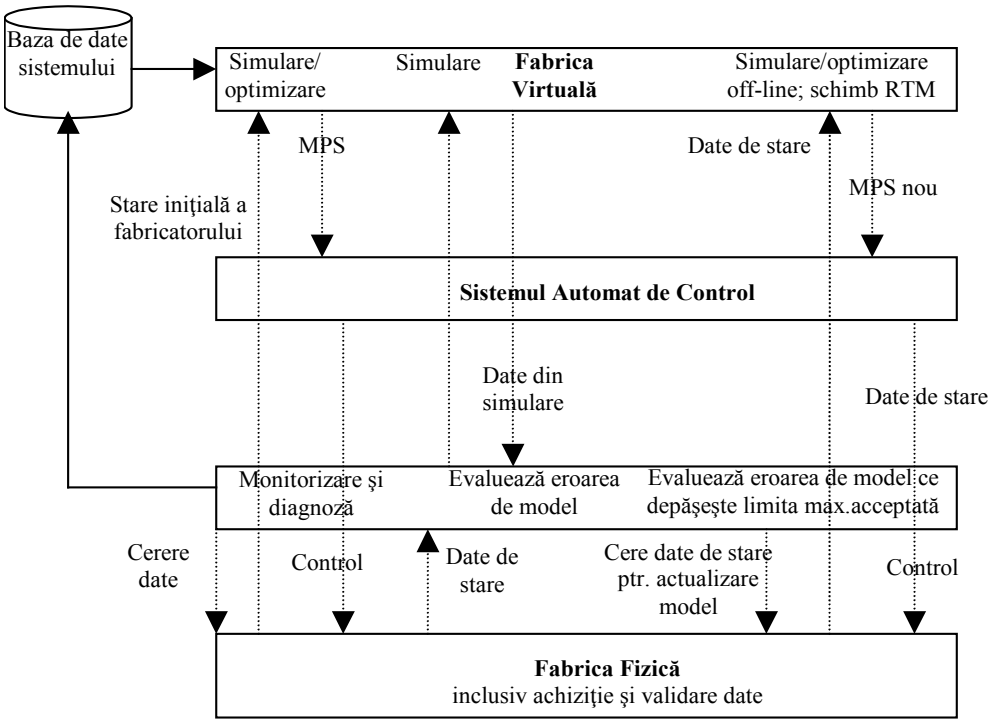


Fig. 4.15.

CONCLUZII

Mediul în schimbare al întreprinderii virtuale, termen propus în acest capitol, deschide o cale către noi domenii ale schimbului de informaţii în spaţiul electronic, inclusiv pentru comerţul electronic, în care partenerii alianţelor virtuale trebuie să acţioneze integrându-se în reţele de companii care schimbă informaţii esenţiale în scopul menţinerii competitivităţii şi a oportunităţii de afaceri.

Definiţia propusă pentru întreprinderea virtuală subliniază că aceasta este o alianţă temporară a unor parteneri focalizaţi pe competenţele lor de bază, care astfel se completează sinergic, alianţă care creează cadrul de cooperare, ca suport al procesului de inovaţie în reţea şi răspuns rapid la cerinţele de afaceri.

Definiţia formală propusă pentru întreprinderea virtuală subliniază faptul că VE este un meta-sistem deschis, care evoluează într-un univers de afaceri şi este formată dintr-un număr de sisteme FABRICATOR, care fac parte dintr-un set fezabil de potenţiali parteneri, care cooperează pentru maximizarea funcţiei obiectiv proprii, în contextul participării la alianţe.

*Arhitectura propusă pentru întreprinderea virtuală este una scalabilă, având la bază module FABRICATOR. Pentru a se asigura compatibilitatea la nivel informaţional a posibililor parteneri într-o alianţă virtuală, se propune o interfaţă neutrală, denumită ,,**Driver de Integrare**" şi se stabilesc specificaţiile funcţionale ale acesteia.*

Pentru a proiecta alianţele virtuale de tip întreprindere virtuală, bazat pe module fabricator compatibilizate la nivel de reprezetare şi schimb de informaţii, se propune un cadru formal şi se dezvoltă un algoritm original.

Validarea unei părţi a algoritmului propus s-a realizat cu ajutorul aplicaţiei KnowledgeNet al companiei americane IMS, construită pentru a putea localiza, în timp real, specialişti având anumite cunoştinţe şi expertiza necesară participării în diferite proiecte sau pentru a putea fi implicaţi ad-hoc în echipe de cercetare multi şi interdisciplinară.

Se propune termenul de reţea de cunoştinţe de reinginerie, utilizând facilităţile Web, prin asocierea dintre o reţea de cunoştinţe despre potenţiali parteneri, fabricator sau parteneriate deja existente, creată pe baza operatorilor definiţi, şi reţea de capitalizare a cunoştinţelor inginereşti.

Este formulată problema bibliotecilor cu acces inteligent, ca suport pentru reţele globale de cunoştinţe de reinginerie. Stabilirea unei Infrastructuri Comune Semantice (ICS) permite întreprinderilor să combine eficient cunoştinţele interne cu cele din exterior, şi să utilizeze experienţa validată a alianţelor virtuale stabilite în timp, prin intermediul unei reţele globale cunoştinţe de reinginerie care să asigure: cadrul semantic pentru sistematizarea cunoştinţelor inginereşti, cadru semantic pentru sistematizarea cunoştinţelor de reinginerie, set de pachete software, o infrastructură necesară capturării, partajării şi întreţinerii

cunoştinţelor specifice. Se propune o arhitectură a sitemului global de cunoştinţe de reinginerie.

Considerând ciclul de viaţă al unei întreprinderi virtuale format din patru faze: faza iniţială evaluarea afacerii şi selectarea partenerilor; faza de configurare şi reconfigurare bazată *pe negociere, reţelarea logistică şi integrarea partenerilor conform algoritmului de proiectare a alianţei virtuale şi pe baza reţelei de cunostinţe de reinginerie; faza de operare ; faza de dezmembrare a alianţei* virtuale se *formuleză problema conducerii sistemelor de fabricaţie integrate în structuri de întreprindere virtuală, legat de asigurarea fazei operaţionale a alianţei virtuale.*

Problema conducerii unei VE este o problemă complexă, de elaborare a unor ,,pachete de decizii" privind operarea ansamblului virtual bazat pe structuri fabricator independente, astfel încât să satisfacă comenzile clienţilor în condiţii de maximixare a funcţiei obiectiv, $O_l^t = f$ (cost, calitate, resurse, timp).

Este analizată problema universului de timp şi a constrângerilor de timp-real din punctul de vedere al unei alianţe VE operaţionale, şi a cerinţelor de timp real necesar a fi îndeplinite în faza de configurare a alianţei, constatându-se că la nivelul formării alianţei virtuale cerinţele de timp real datorate comunicaţiilor în medii electronice evoluate devin comparabile cu cerinţele de timp real la nivelul staţiilor de lucru şi atelier, secunde-minute-ore.

Se propune o arhitectură de planificare şi control, proactivă în raport cu comportarea în cadrul alianţei virtuale şi respectiv reactivă relativ la conducerea sistemului de fabricaţie local.

Arhitectura are la bază agenţi autonomi, propunându-se o configuraţie total descentralizată, specifică întreprinderilor virtuale şi o configuraţie în care partenerii sunt coordonaţi central, corespunzatoare clasificării întreprinderii virtuale funcţie de tipul coordonării.

Capitolul se încheie prin propunerea arhitecturii de control supervizor şi achiziţie de date pentru fabricator, propunându-se utilizarea a trei bucle de control automat: la nivelul echipamentelor, la nivelul modelului de timp real al aplicaţiei şi la modelul de simulare a sistemului de fabricaţie.

Capitolul 5 – cuprins

5.1. Arhitectura de sistem **155**

5.1.1.Laborator pentru cercetări in domeniul fabricaţiei inteligente **159**

- **Platforma hard/software** **161**

5.1.2. Laborator de cercetare pentru SED şi SCADA **175**

5.1.3. Laborator pentru cercetări în domeniul tehnologiei mecanice şi a managementului industrial **176**

Concluzii **178**

5

Studiu de caz: FABRICATOR
Pilot de laborator virtual

5.1. ARHITECTURA DE SISTEM

Realizarea pilotului de laborator virtual de cercetare şi didactic impune definirea unei arhitecturi de referinţă, proiectarea şi dezvoltarea unei platforme, utilizarea unor protocoale de comunicaţii şi a unor mecanisme care să garanteze siguranţa transmiterii informaţiei.

FABRICATORUL este o alianţă de laboratoare creată cu scopul de a utiliza în comun experienţa de cercetare şi cea didactică, având ca ţinte:

- crearea unui mediu virtual de cercetare multidisciplinar şi interdisciplinar, capabil să disemineze rezultate şi cunoştinţe către mediul universitar şi către cel de afaceri (IMM întreprinderi mici şi mijlocii);
- crearea unui nucleu experimental de bază distribuită de cunoştinţe de reinginerie (aşa cum a fost definit în cap.4) în care cooperarea este asigurată printr-un suport IT adecvat.

Conform principiilor de structurare a întreprinderilor virtuale, laboratorul virtual propus este format din trei laboratoare didactice şi de cercetare, având fiecare competenţe de bază specifice, care se completează sinergic.

Arhitectura organizaţiei virtuale propuse este o arhitectură scalabilă, conform figurii 4.1., laboratorul propunându-şi extinderea structurii prin interconectarea altor laboratoare didactice şi de cercetare, prin Internet/Intranet. Interfaţa grafică utilizator (GUI) prezintă un sistem de referinţă spaţial care permite echipelor distribuite geografic să lucreze ca şi cum ar fi în acelaşi loc. Scenariul operaţional de conferinţă desktop, ca în figura 4.2., va permite efectuarea de tele-cursuri, asigurând studenţilor şi cercetătorilor accesul la facilităţile de experimentare ale laboratoarelor partenere.

PLV (prototipul de laborator virtual) este format din următoarele laboratoare, ca nucleu central:

L1: laborator orientat pe fabricaţie inteligentă, modelare şi simulare sisteme hibride, proiectare în mediu multiplatformă de inginerie concurentă (EUCLID,

CATIA, Pro-Engineering), comerţ electronic şi are integrată o platformă de MIS (*Manufacturing Information System*):

- competenţă în proiectare CAD/CAM/CAE;
- pachete de modelare/simulare fabricaţie;
- mediu multiplatformă de IC (inginerie concurentă);
- reţea Jawa pentru comerţ electronic;
- aplicaţie MIS pe platforma IBM-AS400;
- sistem flexibil de fabricaţie orientat pe prelucrare prin strunjire;
- platformă UNIX-like de timp real QNX, pe suport PC;
- reţea Windows NT 4.

L2: laborator orientat pe analiză/proiectare/modelare/simulare sisteme automate de control

- competenţă în analiză/modelare/simulare sisteme cu evenimente discrete;
- platformă de timp real QNX, pe suport PC;
- platformă de timp real OS-9, UNIX-like de timp real, pe suport VME;
- pachete de suport aplicaţii de control supervizor şi achiziţie de date (SCADA) şi nucleu de simulator aplicaţie SCADA;
- platformă de inginerie concurentă CAS-CADE pentru proiectare/modelare /simulare AS.

L3: laborator orientat pe tehnologie mecanică şi management industrial, modelare şi simulare sisteme de fabricaţie şi proiectare off-line de aplicaţii robotizate:

- competenţă în proiectare CAD/CAM;
- pachete de modelare/simulare aplicaţii robotizate şi sisteme de fabricaţie;
- platformă de CAS-CADE de inginerie concurentă.

PLV poate susţine întreg ciclul de viaţă al unui produs mecatronic, iar ţinta cercetărilor viitoare este dezvoltarea unui mediu de realitate virtuală care să permită utilizarea tehnologiei de prototipizare virtuală în mediu de inginerie concurentă, şi dezvoltarea de aplicaţii de control supervizor şi achiziţie de date.

Laboraroarele sunt interconectate la reţeaua Internet, dar se are în vedere crearea unei reţele Intranet pentru a se asigura posibilitatea de comunicaţii în timp real necesară pentru aplicaţii specifice de mare viteză.

Serviciile asigurate sunt grupate în termeni specifici activităţii didactice şi de cercetare şi în termeni de inginerie a afacerilor, astfel:

- ***un mediu didactic şi de cercetare*** orientat pe analiză, modelare, simulare, proiectare şi validare sisteme inteligente de fabricaţie;

Interacţiunea şi partajarea datelor poate avea loc atunci când, în procesul didactic şi de cercetare se desfăşoară o sesiune de prezentare, în care noile produse şi procesele asociate lor sunt „afişate” utilizând tehnici de realitate virtuală şi prototipizare virtuală şi sunt explicate interactiv ca proprietăţi şi utilizarea unor participanţi care pun întrebari şi primesc răspunsuri în timp real.

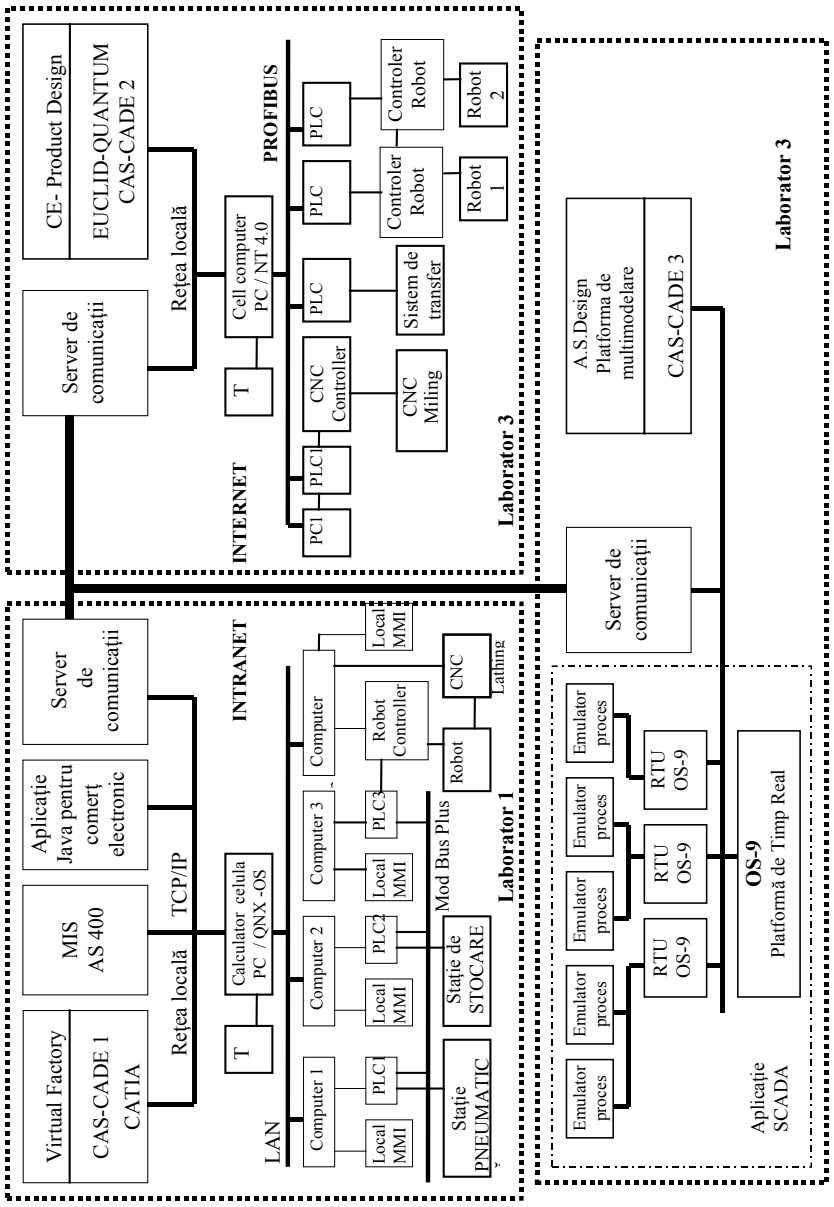


Fig. 5.1. Interconectarea laboratoarelor în FABRICATOR

– ***un mediu de proiectre colaborativă*** pentru produs/proces/sistem

- Interacţiunea şi partajarea datelor se produce când se accesează bazele de date care conţin informaţii despre produs şi părţile sale şi în momentul transferului automat al proiectelor (desene şi specificaţii CAD sau VRML2) între laboratoare, pentru evaluarea produselor într-o sesiune comună de teleconferinţă, pentru analiza din punct de vedere al cerinţelor de fabricaţie, durate de realizare, iar in perspectiva, din punct de vedere al costurilor de realizare şi a selecţionarii de parteneri;
- ***un mediu didactic şi de cercetare în domeniul comerţului electronic***, având şi rol în realizarea alianţelor virtuale pe bază de negociere şi acces la baze distribuite de cunoştinţe de inginerie (cap.4)

La nivelul unui nod, arhitectura propusă este conform figura 5.2.

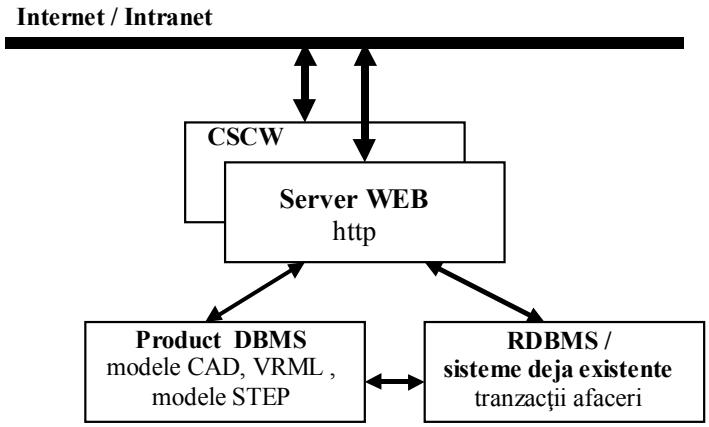


Fig. 5.2. Arhitectura locala a unui nod în PVL

Platformele de integrate alese, SUN-SOLARIS cu soluţii Java pentru server Web (http), CSCW (conferinţe audio-video) şi EC (comerţ electronic) şi CAS-CADE (EUCLID-QUANTUM) pentru mediu de inginerie concurentă asigură, prin funcţiunile specifice, următoarele componente:

- ***mediu colaborativ 3D pentru construirea modelului***; acest lucru este realizat iniţial în VRML 2, software specializat pentru proiectare multi-user interactiv 3D şi Java. Comportarea dinamică şi interacţiunile cu utilizatorul vor fi implementate în Java. Ca mediu de IC se utilizează platforma CAS-CADE EUCLID QUANTUM;
- ***Translator format CAD în format VRML;***
- ***Translator STEP;***
- ***Baza de date orientate pe obiecte*** utilizată pentru memorarea modelelor componentelor, sub-ansamblelor, ansamblelor, în format STEP, CAD şi VRML2.Versiunile modelelor sunt implementate la nivelul componentelor.

Bazele de date tratează în mod transparent datele replicate. Sunt oferite servicii de tip gateway spre alte baze de date similare;

- ***Serverul de Web;*** partenerii accesează informaţii prin Web sau direct prin conectare la baza de date O-O/RDBMS. Clienţii din exteriorul laboratorului virtual au acces numai prin intermediul Web;
- ***CSCW***, include video-coferinţe, audio-conferinţe, white-boarding services etc.

5.1.1. LABORATOR PENTRU CERCETĂRI ÎN DOMENIUL FABRICAŢIEI INTELIGENTE

Arhitectura de sistem

Figura 5.3. prezintă arhitectura de sistem a laboratorului pentru cercetări în domeniul fabricaţiei inteligente.

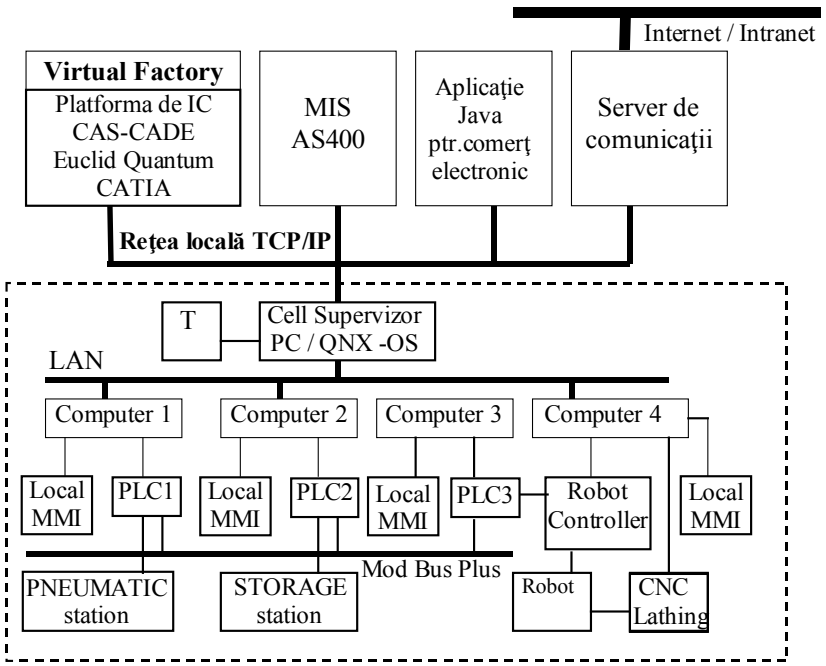


Fig. 5.3. Laborator de cercetări în domeniul fabricaţiei inteligente

Layout-ul sistemului flexibil de fabricaţie este prezentat în figura 5.4.

Fară a intra în detalii privind descompunerea funcţională a sistemului (prima dimensiune a cadrului 4D de multimodelare propus la pct.3.2.) pilot de fabricaţie, se va preciza succesiunea de activităţi realizate într-o funcţionare normală:

- semifabricatele, de formă cilindrică, intră în sistem prin intermediul staţiei pneumatice de alimentare, deservită de R1 şi R2, două manipulatoare/robot pneumatice, ca resurse partajate;
- se face o primă verificare a semifabricatelor utilizându-se un gripper programabil, care poate sesiza diametrul semifabricatelor manipulate;
- semifabricatele sunt depozitate, la intrarea în sistem, în două alimentatoare gravitaţionale.

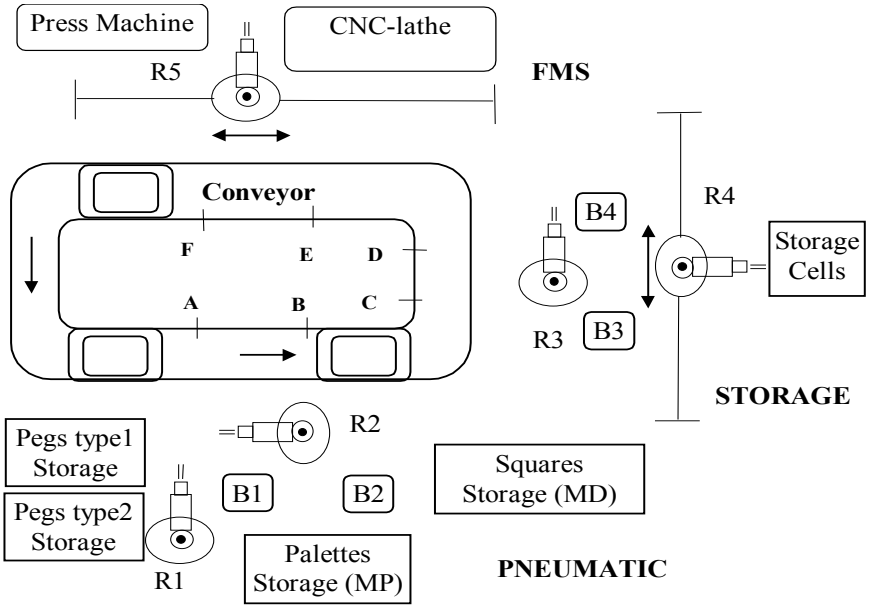


Fig. 5.4. Dispunerea echipamentelor în sistemul flexibil de fabricaţie din L1

- semifabricatele sunt prelucrate de către strung şi asamblate, cilindru-prelucrat cu paralelipiped, prin presare, în staţia de asamblare; cele două maşini sunt deservite de un robot, care joacă rolul de resursă partajată în sistem;
- în situaţia în care FMS-ul nu este disponibil, paletele cu semifabricate sunt stocate în magazie (în figura 5.4 notată cu STORAGE);
- Staţia STORAGE este condusă de un automat programabil şi este deservită de doi manipulatori pneumatici şi un robot electric;
- produsele finite sunt de asemenea depozitate în magazie;

– Pentru recunoşterea paletelor se utilizează un sistem bazat pe senzori inductivi (se utilizează un cod binar) şi un sistem cu cod de bare.

Modelele pe baza formalismului Reţele Petri pentru staţiile din aplicaţia de sistem de fabricaţie L1 au fost dezvoltate utilizând metodologia de proiectare hibridă top-down/bottom-up [Zhou, 95]

Platforma hard/software

Soluţia de sistem de control supervizor şi achiziţie de date

Nucleul de comunicaţii al aplicaţiei este reprezentat de reţeaua Mod BusPlus, reţea de tip inel, realizată pe linie bifilară, care leagă între ele automatele sistemului. Ca o particularitate, reţeaua Mod BusPlus permite cuplarea a 54 de noduri pe o distanţă totală a legăturilor de 450 m. Nodul central este cel care coordonează toată activitatea, punând la dispoziţia celorlalte noduri diferite informaţii privind starea sistemului (avarii, starea magaziei, starea paleţilor de pe conveior etc.).

Al doilea nivel al comunicaţiei îl prezintă Modbus, fiecare nod al reţelei Mod BusPlus se poate lega local printr–o linie serială cu un PC la nivelul căruia se află MMI. De la acest nivel se poate modifica starea curentă a sistemului, în scopul introducerii unor noi condiţii de lucru sau programe.

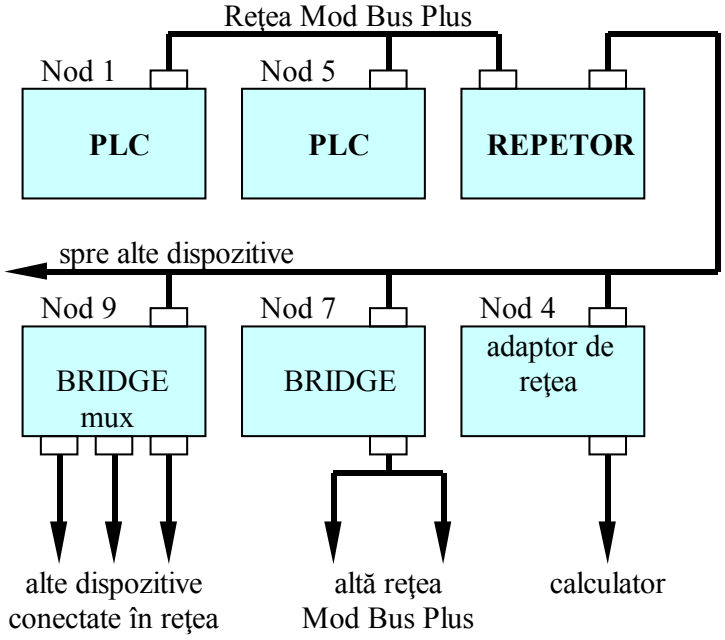


Fig. 5.5. Reţea ModBusPlus

De la nivelul MMI operatorul poate controla, în funcţie de drepturile alocate de administratorul de sistem, toată aplicaţia (automatele fiind legate la reţeaua Mod BusPlus, comenzile date de operator ajung în prima fază în nodul la care este conectat PC–ul, iar apoi, pe reţeaua Mod BusPlus, la automatul aflat în atenţie).
Figura 5.5. arată modul în care se interconecteaza sistemele prin reţea de comunicaţii de tip Modbus.

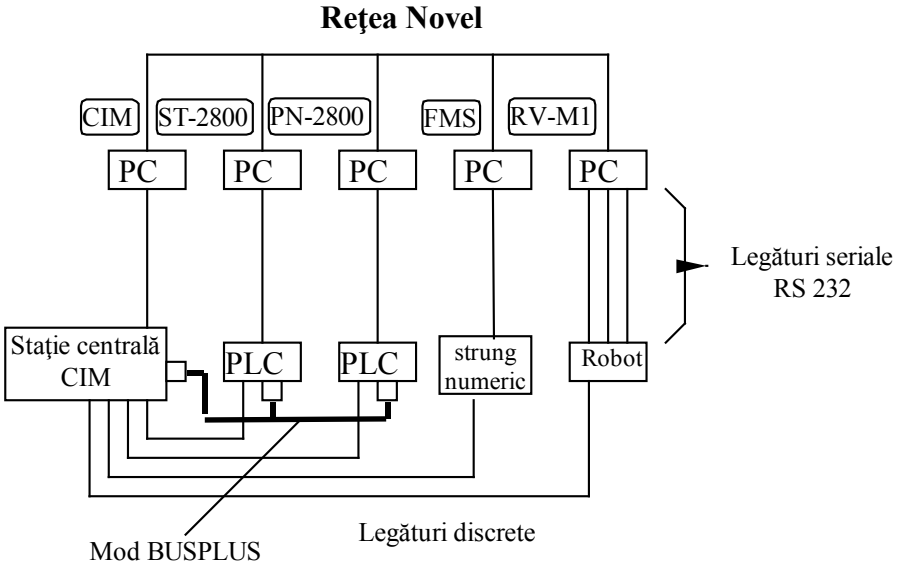


Fig. 5.5. Reţeaua de comunicaţii din L1

Dintre deficienţele majore ale aplicaţiei iniţiale se pot sublinia:

- structură închisă;
- arhitectură soft rigidă;
- pentru studiul unor strategii de conducere a aplicaţiei „Sistem de fabricaţie L1" şi pentru implementarea unui sistem de control supervizor şi achiziţie de date local, trebuie asigurate facilităţi de comunicaţii între calculatorul supervizor şi controlerele locale. La nivelul aplicaţiei iniţiale nu există posibilitatea de a schimba „modelul" de sistem local, aflat la nivelul controlerelor locale (automate programabile AEG) decât într-un regim manual, afectând întrega aplicaţie;
- controlul central este asigurat de un automat programabil clasic (comparativ cu soluţiile moderne de sisteme tip SMART I/O); sistemul este gândit ca un „*macro embeded system*";
- facilităţi minime de modificare a MMI;

- lipsa facilităţilor de timp real; sistem de operare Windows;
- lipsa facilităţilor de utilizare a comunicaţiei în reţea Intranet/Internet.

Soluţia de platformă integrativă experimentală dezvoltată în mediu de Linux, fară facilităţi de timp real

În prima etapă a proiectului de ,,Sistem supervizor şi achiziţie de date pentru L1", 1995-1995 s-a urmărit [Enescu,95] deschiderea aplicaţiei prin utilizarea sistemului de operare Linux, urmând a se realiza trecerea la sistem de operare de tip Unix-like de timp real.

Proiectarea conceptuală a platformei integrative

Structura propusă este structurată în patru niveluri principale, conform figurii 5.7. Cu fiecare nivel creşte gradul de abstractizare a mesajelor din sistem tratate.

AL	**Abstraction Level** oferă o interfaţă unificată şi multiple puncte de acces către nivelurile inferioare
MDL	**Message Dispach Layer** oferă un singur punct de acces spre nivelurile inferioare
IFC	**InterFace Controller** oferă servicii de comunicaţii independente de dispozitiv
LLA	**Low Level Automation** PLC, RI, Smart I/O sau alt controler de proces

Fig. 5.7.

LLA: nivelul acesta nu aparţine strict arhitecturii propuse având în primul rând rolul de a sublinia adaptabilitatea înaltă a nivelului IFC, de a asigura comunicaţia cu mediul fizic, format dintr-o diversitate de echipamente.

IFC: nivelul unifică facilităţile software şi hardware pentru asigurarea diverselor tipuri de interfeţe de comunicaţii:

- oferă nivelurilor superioare pachete standard de date pentru orice tip de controler local ;
- posibilitatea ca senzorii şi efectorii să fie controlaţi de către nivelurile superioare;
- asigură simplificarea comunicaţiei prin stabilirea unui singur punct de acces pentru toate echipamentele, prin utilizarea serviciilor standard de comunicaţii în reţea.

MDL: acest nivel asigură transferul mesajelor între nivelurile superioare şi IFC:
- asigură rutarea mesajelor între diferitele reţele industriale care interconectează echipamentele dintr-o aplicaţie, figura 5.8.;
- asigură punctul de interconectare al AL cu aplicaţia fizică, şi validează pachetele de comandă.

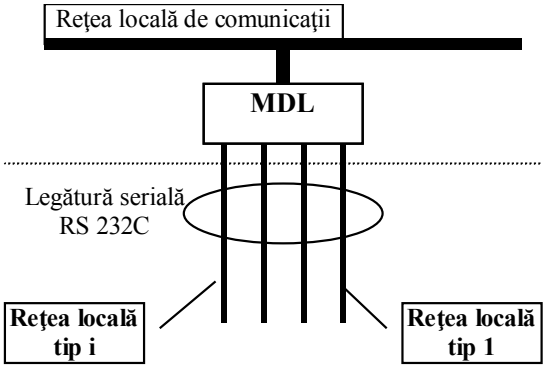


Fig. 5.8.

AL:
- Asigură generarea ordinelor de producţie în secvenţe specifice de comenzi care se trimit controlerelor din system;
- Optimizează ciclii de producţie şi tratează erorile la nivel de system.

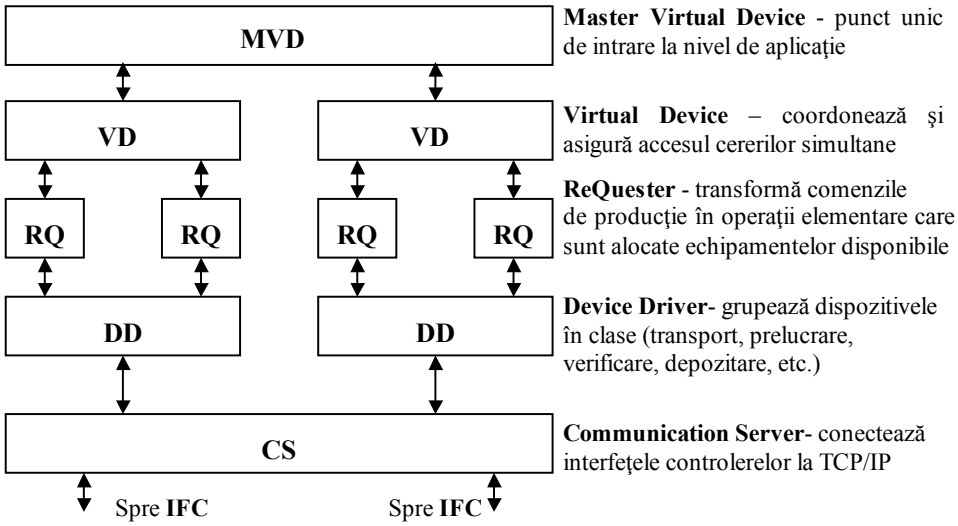


Fig. 5.9. Descrierea modulelor care formează AL

Implementare Device Driver

La implementarea Device Driver-ului (DD) s-a ţinut cont de problemele de comunicaţie cu Requester(R), Virtual Device (VD) şi Comunication Server (CS).

Pentru comunicaţia cu nivelul ierarhic superior s-a propus soluţia de comunicaţie prin **pipe**, iar pentru comunicaţia cu nivelul ierarhic inferior, mecanismul **socket**.

Pentru a realiza izolarea canalelor de comunicaţie între R şi DD s-a ales soluţia alocării dinamice a canalelor de către DD. O dată cu mesajul de răspuns la RQ_R_RESOURCE, DD trimite R şi identificatorul unui canal alocat pentru conversaţie. Structura generală a unui DD este:

```
if ( !plcOK( ) )   {
  ddFatal ( ) ;
}
sendEvONLINE ( ) ;
for ( ; ; ) {
  if (messageVdPresent ( ) )   {
     readVdMessage ( ) ;
     processVdMessage ( ) ;
  }
  if (messageRqPresent ( ) )   {
     readRqMessage ( ) ;
     processRqMessage ( ) ;
  }
  if (resourceAvailable ( ) )   {
     if ( rqPresent ( ) )   {
     processResource ( ) ;
     }
  }
}
```

Din această structură se poate observa o soluţie de implementare a priorităţilor. Întotdeauna un mesaj de la VD este procesat înaintea unuia de la R. Procesul DD este lansat în execuţie de procesul VD corespunzător, după crearea tuturor canalelor de comunicaţii. DD verifică starea automatului şi dacă acesta este on-line trimite mesaj EV-E_ONLINE către VD, după care intră în bucla de procesare a mesajelor.

Implementare Requester

Realizarea Requester necesită rezolvarea problemei rutării dinamice şi a translatării ordinelor de producţie, în primitive acceptate de DD. DD a fost grupat

în trei clase. Routarea dinamică s-a realizat prin înregistrarea ofertelor în baza de date şi menţinerii unei liste cu costurile asociate fiecărei rute.
Structura generală a unui R este:

```
if (!requestPosible ( ) )  {
    rqFatal ( ) ;
}
chooseRoute ( ) ;
processRequest ( ) ;
sendEvStatistic ( ) ;
```

Funcţia „chooseRoute ( )" alege din lista rutelor posibile rută de cost minim. Funcţia „processRequest ( )" analizează cererea şi realizează translatarea efectivă utilizând şi informaţiile din baza de date.

Implementarea Virtual Device

VD trebuie să realizeze înregistrarea în baza de date a ofertei şi dispecerizarea cererilor către R.
Structura unui VD este:

```
getPrimaryInfo ( ) ;
signalMvd ( ) ;
waitMvd ( ) ;
getUpdatedInfo ( ) ;
createChanels ( ) ;
forkDd ( ) ;
for ( ; ; )  {
    if (messageDdPresent ( ) )  {
    processDdMessage ( ) ;
    }
            if (messageRqPresent ( ) )  {
    processRqMessage ( ) ;
    }
            if (messageSockPresent ( ) )  {
    processSockMessage ( ) ;
    }
{
```

unde :

- funcţia „getPrimaryInfo" obţine, din baza de date, informaţii despre oferta proprie şi înregistrează informaţii despre locaţia sa pe reţea;
- funcţia „getUpdatedInfo" obţine din baza de date informaţii despre ofertele celorlalte VD;
- funcţia „processDdMessage" procesează mesajele EV-E_ONLINE şi EV-E_OFFLINE precum şi mesajele de eroare;

- funcţia „processRqMessage” înregistrea EV-E_STATISTIC sau reacţionează la posibilele erori survenite în timpul îndeplinirii ordinului de producţie;
- funcţia „processSockMessage” primeşte cererile de la clienţi, iar pentru fiecare cerere crează un nou R care primeşte acest ordin de producţie.

Implementarea Master Virtual Device

Structura MVD este :

```
getInfo ( ) ;
createVdChanels ( ) ;
forkAllVd ( ) ;
wait AllVd ( ) ;
signalAllVd ( ) ;
for ( ; ; )  {
     if (messageSockPresent ( ) )  {
     processSockMessage ( ) ;
     }
     if (messageVdPresent ( ) )  {
     processVdMessage ( ) ;
     }
}
```

unde :

- funcţia „processSockMessage” răspunde la cererile clienţilor informându-i asupra portului unde pot solicită o anumită piesă;
- funcţia „processVdMessage” înregistrează informaţiile statistice de la VD şi înregistrează erorile ce pot surveni în funcţionarea VD.

În aplicaţia considerată, baza de date conţinând modelul de sistem şi totalitatea informaţiilor parametrice este implementată utilizând conceptul de client-server, care de altfel stă la baza întregii soluţii de sistem.

Dintre concluziile rezultate din această aplicaţie:

- sunt necesare facilităţi de timp real pentru a putea genera aplicaţia SCADA;
- soluţia Unix este cea potrivită, ţinind cont de facilităţile de timp real oferite de soluţii Unix like (comparaţia Unix-like / QNX cu Windows NT şi Windows NT cu facilităţi de timp real constituie subiectul paragrafului următor);
- un test de portabilitate a arătat că, prin compilarea programelor generate sub Linux, pe o maşina IBM RISC 6000, rulând sistemul de operare AIX3.2, au fost necesare modificări numai la părţi ale codului care se ocupă cu gestiunea comunicaţiilor prin socket, ceea ce constituie aproximativ 0.5% din codul generat pentru aplicaţie;

– crearea unui mediu pentru generare de aplicaţii SCADA necesită un efort mare, dificultăţile majore fiind cauzate de multitudinea de drivere specifice către echipamentele fizice.

Soluţia de platformă MMI/SCADA sub OS /ONX 4.2.

Soluţia software care realizează integrarea unei arhitecturi de tip client-server în cadrul L1, precum şi facilităţile de integrare a celulei în atelier, ca nivel ierarhic superior, poate fi reprezentată schematic ca în figura 5.10. Prin interconectarea pe TCP/IP se asigură suportul de comunicaţii pentru integrarea cu celelalte aplicaţii din L1, platformele de CAD\CAM\ CAE, aplicaţia de comerţ electronic şi cu platforma de comunicaţii care asigură interconectarea la Internet.

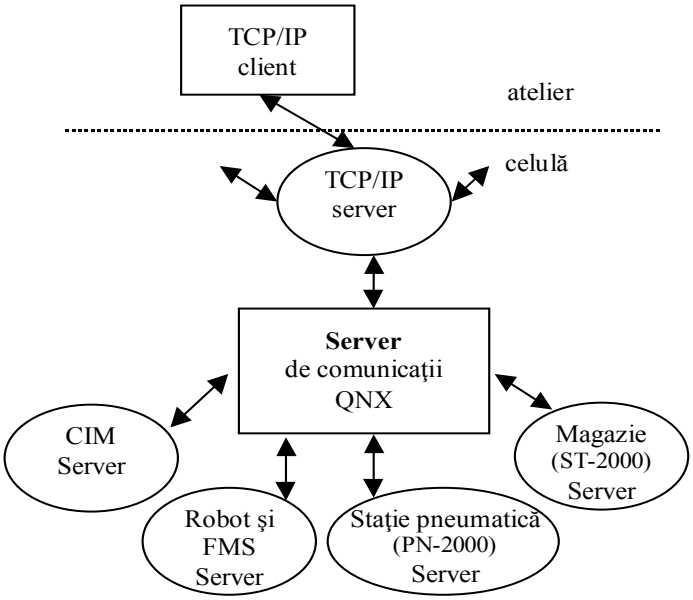


Fig. 5.10.

Clientul şi serverul de TCP/IP sunt modulele care asigură schimbul de informaţii între controlerul de celulă şi controlerul de atelier şi au rolul de a genera, respectiv de a transmite un ordin de fabricaţie (corespunzător modelului de sistem de control al activităţilor de producţie propus în cap.1.).

Serverul de TCP/IP transmite ordinul serverului de comunicaţie QNX, care la rândul său va transmite serverelor staţiilor de lucru programele pe care acestea trebuie să le încarce în controlerele locale.

Serverele staţiilor de lucru confirmă recepţionarea comenzilor.

Serverul şi clientul de TCP/IP

Acest server realizează o interfaţa între reţeaua bazată pe suita de protocoale TCP/IP şi reţeaua QNX. Este un server de tip concurent a cărui funcţionare poate fi reprezentată ca în figura 5.11.

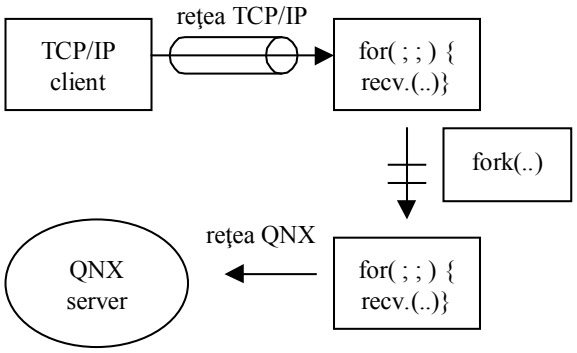


Fig. 5.11

Comunicaţia cu clientul de TCP/IP se realizează prin intermediul socket-ilor.

Serverul creează un socket cu ajutorul directivei *socket()* obţinând un descriptor de socket , după care îi atribuie un nume prin apelul directivei *bind()*. Serverul asteaptă crearea de conexiulni cu clienţii (directivele *listen() şi accept()*).

Clientul de TCP/IP îşi crează şi el descriptor de socket după care solicită conectarea la server, directiva *conect( )*. Schimbul de date se face prin mecanisme *send ()* şi *recv ()*. Comunicaţia cu serverul de QNX se realizează prin mecanismele *send (), receive()* şi *reply()*

Serverul de comunicaţii pentru QNX

Serverul de comunicaţii pentru QNX este cel care dirijează schimbul de mesaje în reţeaua QNX. Imedeiat după lansare înregistrează numele folosind funcţia *qnx_name_attach( )* prin care numele în reţea este stabilit unic, astfel încât reperarea de către clienţi se poate face cu ajutorul funcţiei *qnx_name_locate( )* realizându-se un circuit virtual între două aplicaţii care vor să comunice (nu contează că aplicaţiile rulează în noduri, comunicaţia se desfăşoară ca şi cum ar rula pe acelaşi calculator).

Serverul de QNX este şi el de tip concurent, realizând câte o copie a sa pentru fiecare client care i se adresează. Pentru a " standardiza" comunicaţia, acest server pune la dispoziţia clienţilor o structură de mesaje definită astfel:

```
typedef struct mesaj {
int sender;
int receiver;
char* msg ;
int param;};
```

unde: *sender* - cod asociat expeditorului;
receiver- cod asociat destinatarului;
msg – mesajul propriu-zis;
param – parametru asociat mesajului.

Serverul robot

Este un server de tip iterativ, care la lansare înregistrează numele său în reţea şi setează comunicaţia serială între PC şi controlerul de robot.

Sursele în „C" ale programelor sunt prezentate în [Curaj,98].

Funcţiile de control ale pachetului PCP Virgo

Denumit "Soft-DCS" mediul PCP-Virgo asigură facilităţile necesare asociate cu cele trei niveluri ale MES (*Manufacturing Execution System*):

- **Controlul proceselor –** reglarea, controlul loturilor, controlul secvenţe;
- **Controlul producţiei –** controlul supervizor şi achiziţia de date (SCADA);
- **Managementul de procese –** managementul de baze de date, analiza, inventar, comenzi, calitate, control, optimizare etc.

şi este mediul de dezvoltare utilizat pentru crearea aplicaţiei de SCADA pentru L1. Aceste funcţiuni sunt uzual clasificate conform ierarhiei din figura 5.12.

Cerinţele pentru o aplicaţie software pentru DCS:

- platforma hardware/ software(OS) trebuie să fie standard;
- să facă apel la standarde, de exemplu POSIX, care să asigure interoperabilitatea;
- să poată rula atât pe sisteme ţintă cât şi în reţele extinse;
- să aibă facilităţi de toleranţă la defecţiuni în reţea;
- să admită acces de la distanţă;
- să permită reconfigurarea on-line;
- să aibă facilităţi de timp real.

Modelul de proces (proiectarea orientată pe obiecte asigură facilităţile necesare descompunerii aplicaţiilor complexe de MES în entităţi bine definite, obiecte de nivel înalt, denumite procese) şi modelul ***„unei singure imagini***" (proiectarea într-un mediu distribuit de tip client-server care unifică toate calculatoarele dintr-o reţea, într-un singur supercomputer virtual care leagă nodurile de calcul independente şi autonome într-un tot unitar) sunt concepute astfel încât, corelate, pot conduce la îndeplinirea obiectivelor/cerinţelor formulate anterior.

Modelul de procese; un proces este ***„un sistem de control cuprins într-un sistem de control”,*** o colecţie de resurse care implementează conceptul de automatizare, controlul proceselor şi funcţiile de management procese/resurse care includ:

- I/O şi interfeţe pentru echipamente între aplicaţii;
- Baze de date de timp-real;
- Baze de date SQL;
- Logări;
- Alarme;
- Bucle de reglare;
- Modemuri;
- MMI şi dispozitive pentru interfaţa cu operatorul.

Un proces de tipul propus este echivalentul unui întreg sistem, într-o arhitectură convenţională. Platforma de configurare procese este virtuală, referindu-se la faptul că procesele sunt independente de hardware. Ele pot fi proiectate, testate şi simulate off-line înainte de a fi asociate unei anumite resurse hardware de calcul.

Ca un element autonom în sistem, procesul poate fi alocat sau realocat oricărui calculator din reţea, la orice moment de timp, fără a fi necesară reiniţializarea sistemului şi fără a afecta alte procese. Deoarece un proces continuă să opereze în mod stand-alone în urma unei avarii din reţea, pot fi implementate structuri redundante, scheme de revenire din avarie.

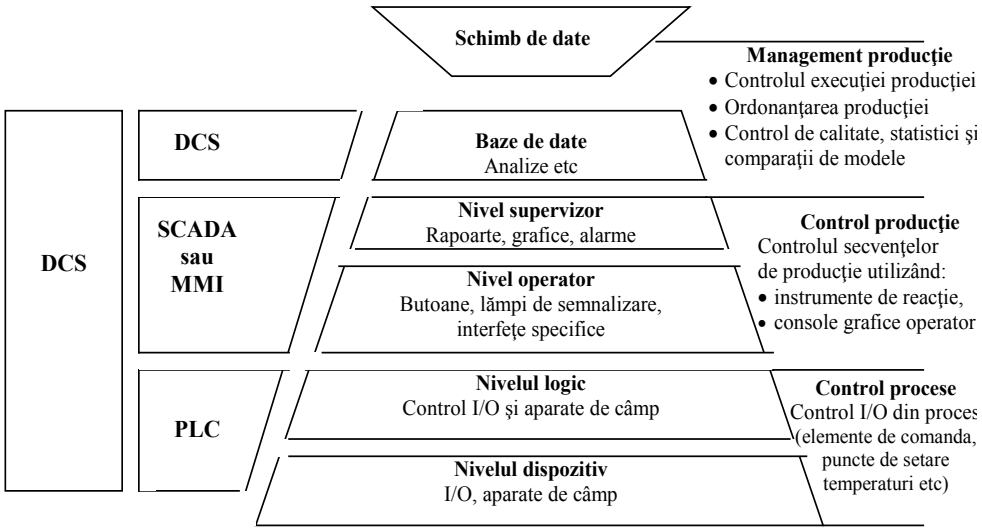


Fig. 5.12. Funcţiile de control ale PCP Virgo/ QNX

Modelul unei singure imagini [Altersys,97]***;*** este baza mecanismului care garantează coerenţa datelor şi permite proceselor să interacţioneze transparent în reţea. Prin acest mecanism se asigură accesul transparent al tuturor proceselor, aflate în oricare nod al reţelei la resurse.

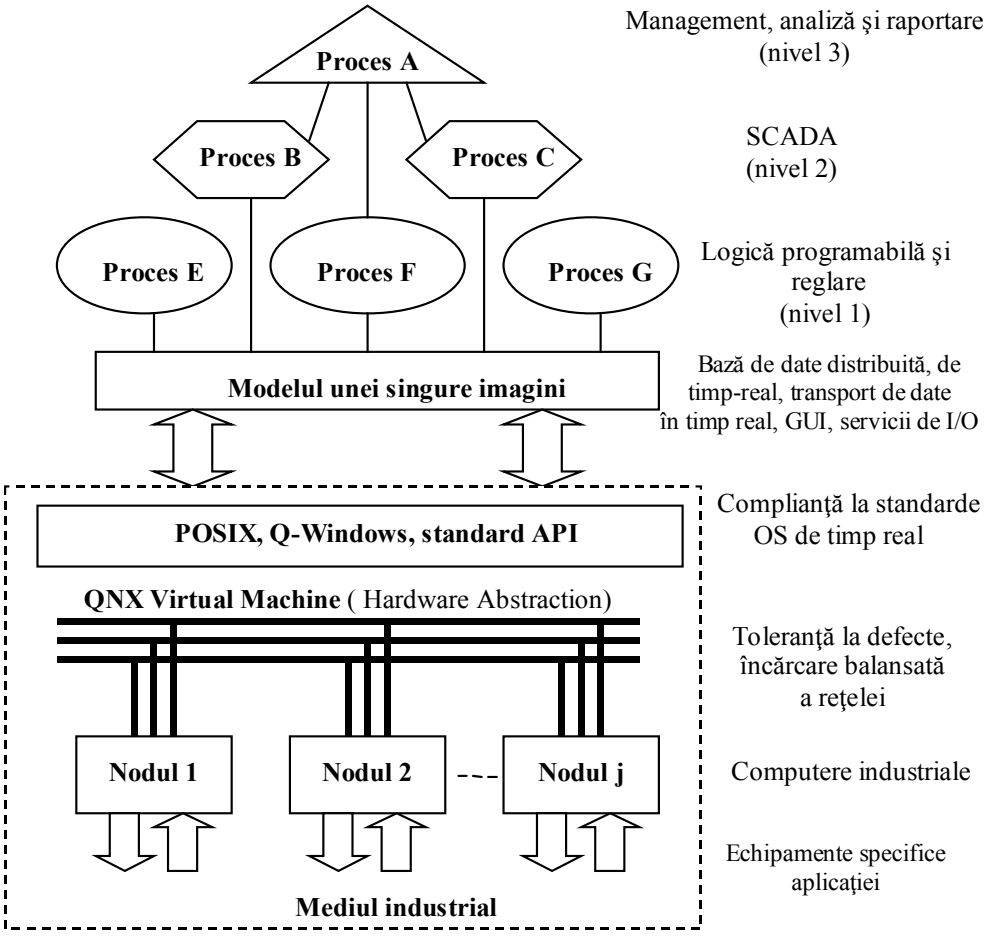


Fig. 5.13. Modelul unei singure imagini

Modelul unei singure imagini îşi asumă taskul referitor la transportul şi menţinerea consistenţei datelor între procese şi de la I/O la procese, în opoziţie cu sistemele clasice DCS, care păstrează copii separate ale datelor de timp real în fiecare nod al arhitecturii, ceea ce generează probleme de coerenţă şi sincronizare.

Alegerea sistemului de operare pentru aplicaţie; QNX vs. NT vs. OS-9

Caracteristici	NT	Extended NT	QNX	OS-9
Procesorul utilizat	Pentium, Pentium Pro, Alpha (necesită un procesor puternic)		80386....Pentium, Pentium Pro, Alpha (scalabil)	Motorola 58000-58030.
Memoria minima necesara	32 MB , 54MB recomandat		8MB - pentru sisteme tinta; 15MB - pentru statia	8M
HDD minim	70MB	10MB - fara POSIX, help, fonts,cursoare	Configuratie posibila fara disk, 4MB disk flash pentru tinte PCP Virgo	Configuratie posibila fara disk
Dimensiune microkernel	>200KB	>200KB	10KB	8KB
Metodele de comunicatii interprocesoare (IPC)	DCOM : tipic 50 msg/sec		10000 msg/s la 10Mb/s Ethernet	-
Multiprocesare	simetrica	simetrica	Simetrica, retea de multi-procesare transparenta pe computere standard	Da
Multi-user	Nu	Nu	Da	Da
Multi-tasking	Da	Da	Da	Da
Multi-hreaded	Da	Da	Da, de la QNX4.24	Da
Driveri retea detasabili	Nu	Nu	Da	Da
OS distribuit	Nu	Nu	Da	Da
Reţea tolerantă la defecte si incărcare balansată	Nu	Nu	Da	Da
Intreţinere de la distanţă	Nu	Nu	Da	Da
Securitate	Certificata C2		Protectii standard UNIX	
API	Win32		Photon	
POSIX API	Da, dar “luata” din Windows API		Da, complet functionala	
Preemptiv	Da		Da	Da
Comutaţie deterministă a task-urilor	Nu, timp de intirziere la intreruperi nespecificat	Da, intirzierea tipică 50µs	Da, intirzierea tipică 5µs	Da, intirzierea tipică 8µs
Aplicaţii de timp real pentru procese rapide	Nu	Da, cu restricţii	Da, performanţe maxime pentru tipul de procesor utilizat	
Scule soft de dezvoltare existente	Da, excelent reprezentate	Da, dar mai puţine decît NT standard	Aplicaţiile standard UNIX, completat cu :	
			Mediul Photon	Mediul MGR

Există un număr de argumente în favoarea lui Windows NT. Acesta beneficiază de un grad de cunoaştere mai mare din partea utilizatorilor şi de asemenea, este instalat pe un număr foarte mare de calculatoare. Totuşi, din punct de vedere tehnic, Windows NT, cu sau fără extensia de timp real, nu asigură performanţele şi capabilităţile solicitate de aplicaţiile de timp real distribuite, din punct de vedere al IPC şi a timpului de răspns, decât cu costuri suplimentare nemotivate practic.

NT nu suportă protocoale de comunicaţii tolerante la defecte şi nici protocoale care să asigure încărcarea balansată a reţelei de comunicaţii.

Din punct de vedere al vitezei, soluţiile cele mai rapide sunt totuşi de 10-100 ori mai lente decât QNX.

Concluzia este că Windows NT nu este potrivit pentru sisteme rapide de timp real, dar poate fi utilizat cu rezultate bune în aplicaţii fără cerinţe stricte de timp real, în sisteme în care există partiţie de timp real bazată pe QNX şi o partiţie NT.

Soluţia propusă este aceea de a utiliza NT, cu toate facilităţile Windows-ului, pentru configurarea şi funcţionarea interfeţei utilizator (Windows-ul rulând ca aplicaţie QNX), în timp ce funcţiile de timp real sunt alocate QNX-ului.

O comparaţie între domeniile de aplicabilitate ale QNX şi OS-9 subliniază că, facilităţile QNX au fost puternic dezvoltate pentru aplicaţii de control distribuit, suportul pe care lucrează este relativ ieftin (platforme compatibile PC) şi poate acoperi aplicaţii diverse la costuri relativ reduse, în timp ce OS-9, asociat standardului VME şi procesoarelor Motorola asigură posibilitatea realizării unor aplicaţii de timp-real, sisteme ţintă de mare performanţă ca viteză, gamă de temperaturi, multiprocesare, la costuri însă semnificativ mai mari, pe echipamente gândite în standard industrial sau militar.

Soluţia hard/software pentru dezvoltare produs utilizată în cadrul L1 oferă un mediu multiplatformă, care asigură atât condiţiile experimentale locale pentru studiul interoperabilităţii modelelor de produs generate, cât şi suportul unui mediu de inginerie concurentă, pentru proiectare produse şi procese.

Mediul multiplatformă este organizat astfel:

- CAS-CADE /***Quantum-Euclid*** pe suport hardware staţie Sylicon Graphics ;
- ***CATIA,*** pe suport staţie RISC IBM 5000 ;
- ***ProEngineering,*** pe suport PC.

Fişierele standard STEP generate asigură schimbul de informaţii între platformele locale şi transmiterea acestora către L2 şi L3.

Utilizarea aceloraşi platforme de inginerie concurentă în toate laboratoarele, CAS-CADE / EUCLID Quantum asigură interoperabilitatea modelelor şi se costituie ca prim suport pentru un nucleu de reţea locală de cunoştinţe inginereşti.

Serverul de comunicaţii Internet/Intranet, o platformă SUN/SOLARIS, asigură facilităţile de integrare a L1 în reţea şi protecţia datelor, în momentul de faţă preluând, cu facilităţi minime, rolul de driver de integrare, propus în cap.4.

5.1.2. LABORATOR DE CERCETARE PENTRU SED şi SCADA

Arhitectura de sistem

Figura 5.14. prezintă arhitectura de sistem a laboratorului pentru cercetări în domeniul SED şi SCADA**.**

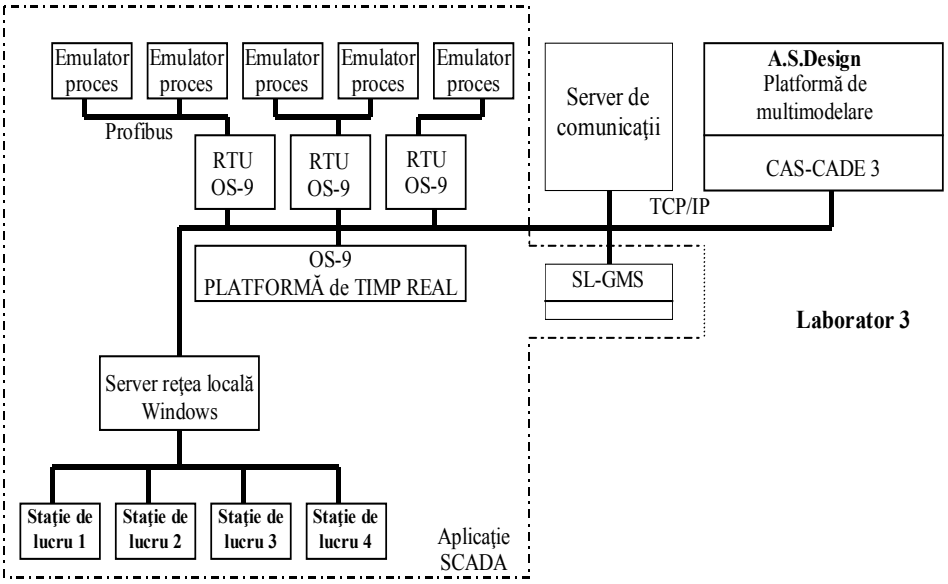


Fig. 5.14. Laboratorul L2

Platforme hard/software

Din punctul de vedere al arhitecturilor de control supervizor şi achiziţie de date, laboratorul L1 utilizează ca suport hardware automate programabile de tip AEG, pentru rezolvarea locală a problemei de conducere a staţiilor de lucru şi calculatoare PC pentru MMI (simulări, configurări, control manual, afişare stare etc.), iar ca soluţie software, QNX / Windows NT.

Soluţia hard/software pentru L2 oferă un mediu multiplatformă, care asigură atât condiţiile experimentale locale pentru cercetare şi cu scop didactic de modelare şi simulare SED, dar şi utilizarea acestor modele în sisteme SCADA simulate în mediu virtual, utilizând modele virtuale (în sensul definit în cap.3) realizate prin simulare în mediul de timp real VME /OS-9 sau Smart I/O /OS-9.

De asemenea, se asigură posibilitatea utilizării facilităţilor de timp real oferite de platforma VME /OS-9, ca o contribuţie la dezvoltarea de aplicaţii SCADA

pentru procese rapide cât şi pentru abordarea soluţiilor de tip SCADA prin prisma sinergiei oferite de platformele PC/NT, PC/QNX, VME/OS-9.

Se asigură integrarea L2 mediului distribuit pentru studiul interoperabilităţii modelelor generate, cât şi pentru generarea reţelei de cunoştinţe de reinginerie.

Mediul multiplatformă din L2 este organizat astfel:

- CAS-CADE /*Quantum-Euclid* pe suport hardware staţie Sylicon Graphics
- CAS-CADE / SL-GMS pentru SCADA ;
- PC/NT cu pachetul SCADA Lookout (*National Instruments*) ;
- reţea locală de PC-uri (staţii de lucru) ;
- reţea locală formată dintr-un sistem VME şi un număr de sisteme tip SmartI/O *(Smart I/O sunt sisteme de timp real OS-9, care realizează funcţiile unui automat programabil ca task al sistemului de timp real, existând resursele în sistem pentru definirea şi lansarea altor taskuri prin care să se simuleze comportarea unor procese de tipul celor din fabrica fizică în aplicaţii locale sau SCADA) ;*
- reţea de platforme SUN /SOLARIS care asigură ;
- facilităţile de integrare a L2 în reţeaua de comunicaţii, protecţia datelor, în momentul de faţă preluând, cu facilităţi minime rolul de driver de integrare, propus în cap.4 ;
- suportul pentru programe de modelare /simulare SED şi simulatoare pentru sisteme de fabricaţie;

5.1.3. LABORATOR PENTRU CERCETĂRI ÎN DOMENIUL TEHNOLOGIEI MECANICE ŞI AL MANAGEMENTULUI INDUSTRIAL

Arhitectura de sistem

Figura 5.15. prezintă arhitectura de sistem a laboratorului pentru cercetări în domeniul SED şi SCADA.

Platforme hard/software

Soluţia hard/soft pentru L3 oferă un mediu multiplatformă, care asigură atât condiţiile experimentale locale pentru cercetare şi cu scop didactic pentru modelarea şi simularea sistemelor de fabricaţie şi a aplicaţiilor robotizate, cât şi platforma de inginerie concurentă care să asigure suportul creării unei reţele de cunoştinţe inginereşti între cele trei laboratoare.

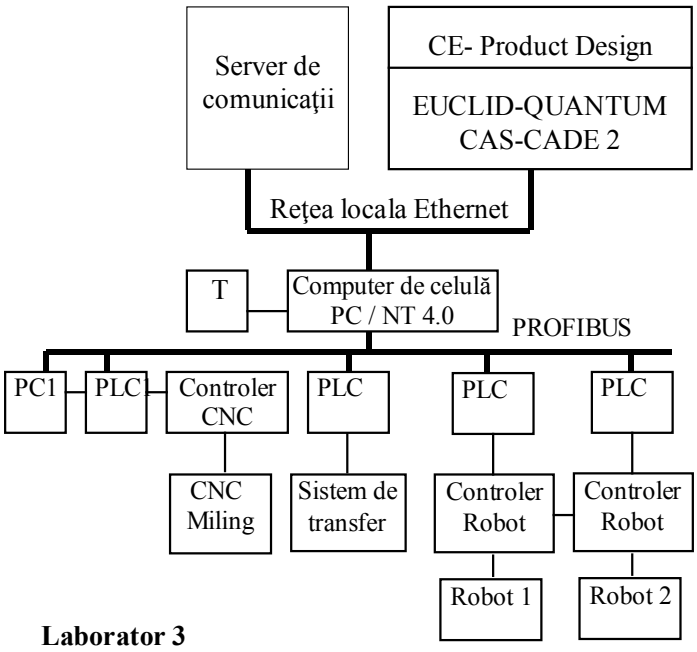


Fig. 5.15.

Mediul multiplatformă din L3 este organizat astfel:

- CAS-CADE/***Quantum-Euclid*** pe suport hardware staţie Sylicon Graphics;
- reţea locală de PC-uri;
- platforma SUN/SOLARIS care asigură integrare a L3 în reţeaua de comunicaţii, protecţia datelor, în momentul de faţă preluând, cu facilităţi minime, rolul de driver de integrare, propus în cap.4;
- suportul pentru programe de modelare/simulare sisteme de fabricaţie (TAYLOR, SIMAN-CINEMA, SIMFACTORY).

CONCLUZII

Soluţia de sistem propusă pentru pilotul de ,,întreprindere virtuală" are rolul de a crea în premieră locală un laborator virtual, având atât scop didacti, cât şi de cercetare în domeniul fabricatorului, fabricii virtuale şi a întreprinderii virtuale.

Laboratorul virtual astfel creat are ca scop:

- *dezvoltarea de aplicaţii de control supervizor şi achiziţie de date prin studierea diverselor soluţii hard/software şi a diverselor soluţii de mediu de dezvoltare de timp real;*
- *crearea unui mediu multiplatformă pentru dezvoltare produs/proces în mediul de inginerie concurentă şi studiul interoperabilităţii dintre model;*
- *crearea unui mediu de modelare/ simulare pentru studiul SED;*
- *crearea unei infrastructuri de comunicaţii care să asigure, pentru început, o parte a funcţiilor driverului de integrare propus în cap.4, cât şi mediul pentru validarea/dezvoltarea unui pilot de reţea de cunoştinţe de reinginerie, propusă în cap.4;*
- *crearea unui suport tehnic compatibil care să asigure integrarea laboratorului virtual în reţeaua Europeană de Inginerie Concurentă;*
- *crearea unui suport pentru aplicaţii de comert electronic în care să se poată studia cerinţe de compatibilitate, soluţii de interoperabilitate privind modelele de produs asociate unei comenzi efectuate în format EDIFACT, tehnici de protecţie la accesul neautorizat.*

Soluţiile tehnice propuse se bazează pe o experienţă acumulată de-a lungul unei periode de câtiva ani în care autorul a studiat aplicarea sistemelor de timp real pentru conducerea proceselor de fabricaţie şi a roboţilor industriali, soluţii de interconectare a echipamentelor de comandă de tip automat programabil, calculator, calculator de proces prin standarde diverse de comunicaţii şi, de asemene, se bazează pe specificaţiile formulate de reţeaua europeană de inginerie concurentă în ceea ce priveşte compatibilitatea hardware şi software la nivel de facilităţi de comunicaţii şi medii de dezvoltare produs/proces/sistem.

ANEXE

A1. PROBLEMA CONDUCERII SED: SUPERVIZARE ŞI CONTROLABILITATE

Problema sintezei unui controler pentru SED a fost tratată iniţial în lucrările lui Ramadge şi Wonham, care au făcut primele cercetări asupra condiţiilor de existenţă şi tehnicilor de sinteză a acestora [Ramadge,82], [Ramadge,89]. În aceste lucrări a fost introdus termenul de *supervisory control* – conducere prin supervizare, care este însă folosit şi sub forma contrasă de **supervizare**.

În continuare sunt prezentate ideile principale din teoria supervizării SED, conform [Caramihai,97]. În abordarea Ramadge-Wonham studiul sistemelor este exclusiv calitativ – timpul nu intervine deloc – procesul este descris prin intermediul automatelor de stare, deci pe baza unui model logic. Procesul este considerat ca fiind un sistem SED care evoluează spontan, generând evenimente. **Supervizorul** este şi el un SED care este cuplat cu procesul astfel, încât intrările sale sunt ieşirile procesului şi reciproc. Supervizorul poate modifica funcţionarea procesului autorizând sau interzicând apariţia evenimentelor în proces.

Dându-se un proces şi un ansamblu de specificaţii logice de funcţionare pentru acesta, obiectivul teoriei Ramadge-Wonham este de a sintetiza un supervizor astfel încât funcţionarea ansamblului proces-supervizor să respecte specificaţiile. În plus, se doreşte ca funcţionarea astfel obţinută să fie cea mai permisivă posibil. Sinteza unui astfel de supervizor se bazează pe conceptul de **controlabilitate**.

Ipoteza de lucru este cea a *observabiliţătii totale* – respectiv toate evenimentele produse de către proces sunt observabile de către supervizor.

În cazul reprezentat în figura A.1., există un singur supervizor cuplat la proces. Acest tip de supervizare se numeste *centralizată*. Atunci când procesul este cuplat la mai multe supervizoare, se poate vorbi de supervizarea *modulară*.

Principiul supervizării modulare

Procesul este considerat a fi un SED care evoluează în mod spontan generând evenimente; un astfel de sistem este denumit *generator de evenimente*. Funcţionarea sa poate fi descrisă printr-un ansamblu de secvenţe de evenimente care constituie un limbaj formal peste alfabetul de evenimente respectiv. Dându-se un astfel de generator G, aflat în starea q, mulţimea evenimentelor care pot fi generate spontan de catre G pornind de la starea q este reprezentată de alfabetul notat $\Sigma_G(q)$.

Un proces cuplat la un supervizor poate fi considerat ca un sistem care primeşte la intrare o listă de evenimente interzise Ω. Acest proces, notat G(Ω) se numeşte *proces supervizat* (figura A.1.).
Procesul supervizat poate genera un eveniment σ dacă σ poate fi generat de procesul izolat şi dacă $\sigma \notin \Omega$.
În figura A.1 procesul este în stare q. Din această stare, $G(\Omega^{(i)})$ poate genera, la momentul (i+1), evenimentul $\sigma^{(i+1)}$ care este un element al mulţimii $\Sigma_G(q) \setminus \Omega^{(i)}$. Supervizorul observă procesul în mod asincron. Apariţia evenimentului $\sigma^{(i+1)}$ poate să conducă supervizorul într-o nouă stare, în care el va genera o nouă listă de evenimente interzise, şi aşa mai departe.

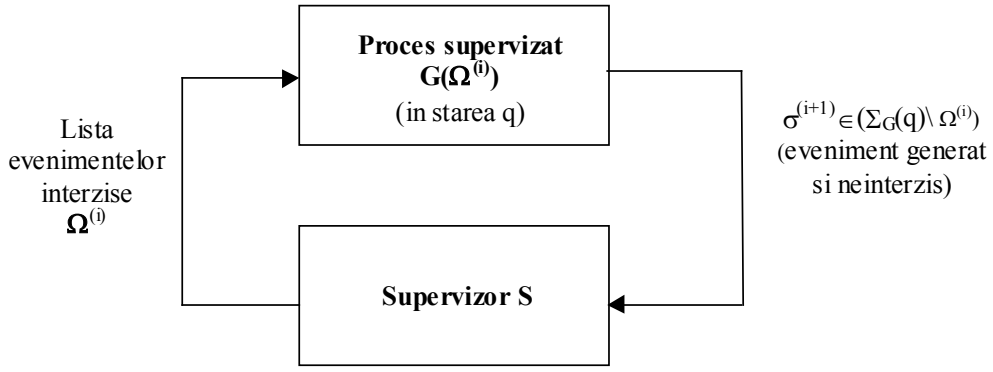


Fig. A.1.

Funcţionarea procesului cuplat la supervizor se numeşte *funcţionare în buclă închisă*.

Observaţie: Un proces supervizat se poate defini echivalent dacă se specifică lista de evenimente autorizate. Dacă Σ este alfabetul de evenimente al procesului, atunci lista evenimentelor autorizate este $\Sigma \setminus \Omega$.

Observaţie: Rolul supervizorului este doar de a interzice apariţia unor evenimente. În nici un caz el nu poate forţa apariţia de evenimente. Prin urmare, efectul supervizarii este de a *restricţiona* comportarea procesului.

În realitate, anumite evenimente nu pot fi în nici un fel impiedicate să apară (cazul defecţiunilor de echipamente în SFF). Un astfel de eveniment se numeşte **necontrolabil.** ***Un evenimet care poate fi interzis se numeşte*** **controlabil.**
Dacă Σ este alfabetul evenimentelor generate de către un proces, atunci se pot defini submulţimile disjuncte ale acestuia Σ_u – mulţimea evenimentelor

necontrolabile şi Σ_c – mulţimea evenimentelor controlabile. Lista evenimentelor interzise nu poate să includă nici un element al mulţimii Σ_u: $\Omega^{(i)} \cap \Sigma_u = \Theta$.

Ipoteză de lucru: Procesul nu poate genera două evenimente simultan.
Această ipoteză, care se bazează pe faptul că procesul evoluează într-un timp continuu, pe acela ca supervizorul îl observă într-o manieră asincronă şi pe ipoteza că durata unui eveniment este nulă, permite garantarea faptului că, după apariţia unui eveniment oarecare, supervizorul are timp să evolueze şi să genereze noua listă de evenimente interzise.

Dacă se notează cu L(G) limbajul generat de către proces, cu L(S) cel generat de către supervizor şi cu L(S/G) limbajul care caracterizează funcţionarea în buclă închisă a procesului (limbajul generat de procesul supervizat), atunci (figura A.2.)

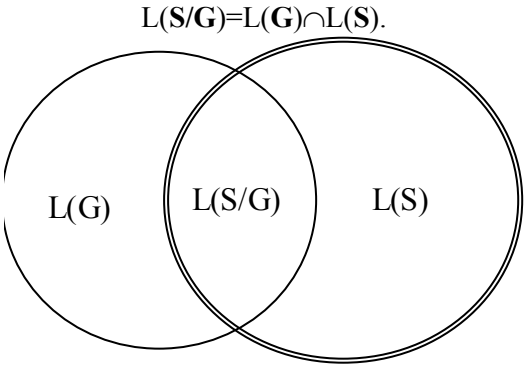


Fig. A.2.

Conceptul de controlabilitate
Fie un proces G şi o specificaţie de funcţionare Spec. Se doreşte sinteza unui supervizor S, astfel încât sistemul în buclă inchisă S/G să funcţioneze respectând specificaţia. În termeni de limbaje, dacă L(Spec) este limbajul specificaţiei, modelată ca SED, sinteza supervizorului revine la respectarea condiţiilor

$$L(S/G) \subset L(G) \quad ; \text{prin definiţie}$$
$$L(S/G) \subset L(Spec)$$

Astfel, dacă se doreşte obţinerea funcţionării în buclă închisă celei mai permisive posibil, în cel mai bun caz avem

$$L(S/G) = L(G) \cap L(Spec) = L_D$$

,unde L_D se numeşte *funcţionare dorită*.

Existenţa lui L_D este dependentă de noţiunea de *controlabilitate*.

Definiţie: Un limbaj $K \subset \sum$ este numit ***controlabil*** (în raport cu L(**G**) şi $\sum_u$) dacă $\overline{K}\sum_u \cap L(\mathbf{G}) \subseteq \overline{K}$ - adică este întotdeauna posibil, prin interzicerea execuţiei tranziţiilor controlabile, să se restrangă L(**G**) la K.

În această definiţie K se interpretează ca L(Spec) şi se spune că *limbajul specificaţiei este controlabil în raport cu limbajul procesului.*

Următoarele două teoreme, enunţate de Ramadge şi Wonham dau *condiţiile necesare şi suficiente de existenţă pentru un supervizor.*

Teorema 1: Pentru un limbaj nevid $L \subseteq L(\mathbf{G})$ există un supervizor **S**, astfel încât L(**S/G**)= L dacă şi numai dacă L este *prefix-closed* şi controlabil.

Teorema 2*:* Pentru un limbaj nevid $L \subseteq L_m(\mathbf{G})$ există un supervizor **S**, astfel încât $L_m(\mathbf{S/G}) = L$ dacă şi numai dacă L este $L_m(\mathbf{G})$ - *closed* şi controlabil.

A2. SOLUŢIE DE SISTEM PILOT SCADA, PENTRU L2.

Sistemul oferă următoarele facilităţi:
- controlul şi comanda utilajelor;
- urmărirea parametrilor tehnologici;
- urmărirea consumurilor specifice de materie primă;
- urmărirea şi stocarea informatiei de producţie;
- urmărirea şi înregistrarea alarmelor/avariilor;
- stocarea informaţiilor pentru mentenanţă;
- pregăteşte informaţia pentru sistemul de gestiune economică.

Sistemul de control este realizat cu:
- echipamente de înaltă tehnologie bazate pe VME bus;
- arhitectură modulară;
- software de timp real (OS-9).

Descriere tehnică
"Fabricatorul L2 " este împărţit în "**zone tehnologice"**, iar fiecare zonă tehnologică are alocate mai multe "utilaje şi instalaţii tehnologice", fiecare cu un anumit număr de mărimi de intrare/ieşire, simulate prin sisteme SMART I/O.
Sistemul de control al aplicaţie pilot este:
- distribuit pe zone;
- interconectat pe reţele;
- ierarhizat.

Elementele sistemul pentru conducerea proceselor sunt controlerul de zonă şi supervizorul.

Controler-ul de zonă (ZC) este un echipament, de sine stătător, conduce procesul tehnologic la care este ataşat. Funcţiile specifice procesului la care este conectat ZC sunt implementate în acesta prin intermediul ISaGRAF - software pentru programarea controler-ului de zonă ZC.
Controler-ul de zonă (ZC) se cuplează la proces şi realizează:
- controlul procesului din zona tehnologică aferentă, pe baza algoritmului implementat cu ISaGRAF;
- memorarea operaţiilor realizate;
- pornire/oprire;
- stocare timpi de funcţionare;
- comunicaţia de date şi comenzi între controlere aferente zonelor; tehnologice şi între controler şi supervizor.

Supervizorul: este conectat la magistrala de câmp şi oferă funcţii de:
- control şi supervizare a funcţionării întregului proces tehnologic;
- coordonare a întregului sistem;
- monitorizarea proceselor;
- rutine de alarmă şi de tratare a situaţiilor de avarie;
- management informaţional;
- rapoarte;
- stocare informaţii de proces pe termen lung;
- calcule statistice şi de gestiune economică.

Arhitectura sistemului

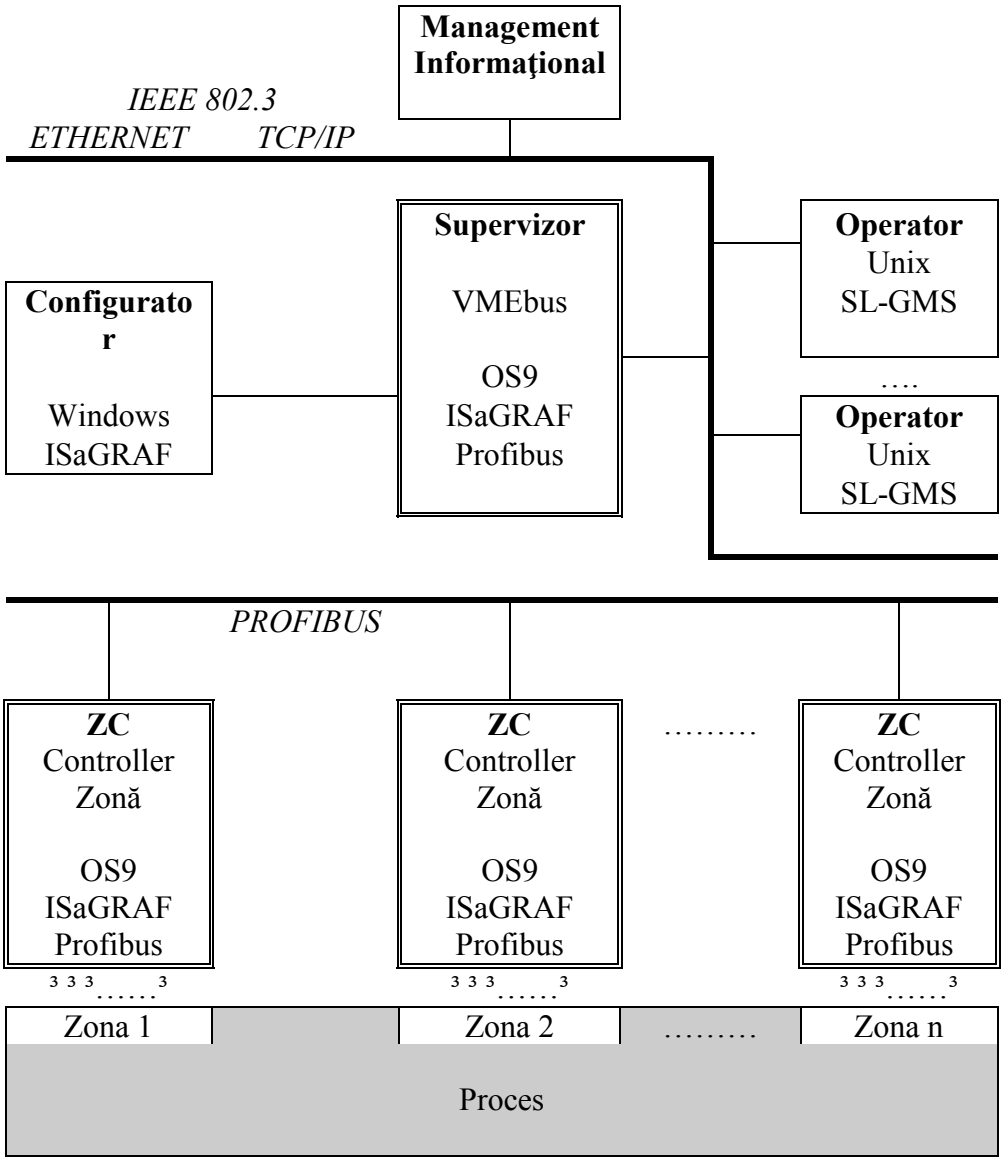
Sistemul pilot are o arhitectură distribuită, gândită pe niveluri ierarhice şi cu posibiliţăti de extindere. Arhitectura se bazează pe o reţea de proces (PROFIBUS), care conectează între ele echipamentele cuplate la proces, precum şi nivelul ierarhic superior. Prin intermediul acestei magistrale de proces se pot comunica date şi comenzi pentru conducerea optimă a proceselor industriale, simulate în laborator.
Supervizorul sistemului se conectează la reţeaua locală de calculatoare prin TCP/IP şi furnizează date în formatul necesar sistemului de management informaţional.

Descriere componente sistem (Hardware şi Software)

Nivelul PROCES

Sistemul de control se conectează la proces prin intermediul unităţilor denumite Zone Controler (ZC). Un ZC furnizează o gamă completă de capabilităţi de achiziţie de date şi control, incluzând intrări şi ieşiri digitale, intrări şi ieşiri analogice. Atât funcţiile de control, care sunt implementate pe ZC, cât şi numărul şi tipul I/O se configurează în funcţie de aplicaţia concretă. Pentru a avea posibilitatea de a comanda procesul şi de la o consolă locală, nu numai de la nivelul superior, la ZC se poate conecta serial un panou operator cu tastatură şi display alfanumeric.
Hardware

Un ZC este realizat un echipament bazat pe o unitate centrală şi un sistem de intrări/ieşiri, adaptat condiţiilor industriale, cu fiabilitate ridicată, imunitate mare la zgomote şi la alţi factori perturbatori. Elementele principale ale Controller-ului de Zonă sunt: unitate centrală, memorie pentru programare (EPROM), memorie de lucru (RAM), memorie pentru reţinere date la pierderea alimentării de la reţea (SRAM), controller de intrări/ieşiri, unitate pentru comunicaţie serială, sistem de operare



Software: Sistem de operare OS-9

Software pentru programarea ZC – **ISaGRAF care asigură facilităţi de s**imulare off-line, de programare *on-line* şi suportă protocolul *PROFIBUS DIN 19245*. Peste sistemul de operare este instalat *kernel-ul* de ISaGRAF, care permite dezvoltarea şi implementarea de aplicaţii descrise în ISaGRAF, folosind un calculator PC.

Comunicaţie - PROFIBUS:

Profibus este o magistrală de câmp cu standard deschis (EN 50170) care permite dispozitivelor de la diverşi vânzători să comunice fără a adaugă interfeţe de adaptare adiţionale. Este o soluţie de cost scăzut disponibilă pe variate platforme hardware. PROFIBUS oferă o arhitectură software puternică: Standard european EN 50170,DP, FMS şi PA; Soluţii Microcode şi ASIC; Integrat în sistemele de operare în timp real; asigură comunicaţii de înaltă viteză (12 Mbauds) .

Soluţia PROFIBUS este importantă şi prin prisma faptului că este soluţia de standard de comunicaţii şi în Fabricatorul-L3, ceea ce oferă posibilitatea unui mediu distribuit interconectat prin acest standard.

Nivelul SUPERVIZOR

Supervizorul este o staţie de urmărire, coordonare şi control cu interfaţă grafică om / maşină. Aceasta oferă o fereastră asupra întregului sistem, indiferent dacă datele provin din staţia de urmărire şi control, sau sunt residente în Controlerele de Zonă cuplate la simulatoarele de proces.

Hardware: Sistem VME: unitate centrală, interfeţe seriale, interfaţă Profibus, interfaţă reţea. Staţii de operare: display, unitate de stocare informaţii, interfaţă reţea, imprimantă. Consolă de configurare : calculator tip PC/AT

Software :OS-9, ISaGRAF, PROFIBUS

- Software de aplicaţie: SL-GMS cu funcţii de: achiziţie de date, comandă, procesare alarme, procesare evenimente, urmărire, calcule, comunicaţie, display

Pentru testele preliminare de comunicaţie între VME/OS-9 şi staţia SyliconGraphics /SL-GMS s-a utilizat programul :

```
/* program de comunicaţie TCP/IP */
/* includes */
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <modes.h>
#include <types.h>
#include <memory.h>
#include <INET/socket.h>
#include <INET/în.h>
#include <INET/netdb.h>
/* defines */
#define SOCKET_PORT     1500
/* forward declarations */
/* typedefs */
/* globals */
```

```
void main (int argc, char **argv )
{
      struct  date {
            char canal[6];
            char analog_input[20];
            char analog_output[20];
            char digital_input[32];
            char digital_output[32];
                        };
struct date nivel;
struct sockaddr_în server, to;
int sx, s, size, i;
char nr_canal[0]={"0"};
/* deschidere socket TCP/IP */
if ((sx=socket(AF_INET,SOCK_STREAM,0)) == -1)
{
      fprintf(stderr,"Nu pot deschide socket-ul
\n",errno);
}
/* bind the socket to SOCKET_PORT */
server.sin_family = AF_INET;
server.sin_port = htons(SOCKET_PORT);
server.sin_addr.s_addr =INADDR_ANY;
if(bind(sx,(struct sockaddr
*)&server,sizeof(server))==-1)
{
      close(sx);
      exit(_errmsg(errno,"bind failed - "));
}
/* aşteaptă o conexiune */
if (listen(sx, 1) <0)
{
      fprintf(stderr,"tcp_listen - failed\n");
      close(sx);
      exit(errno);
}
/* acceptă conexiunea */
size=sizeof(struct sockaddr_în);
if ((s=accept(sx, (struct sockaddr *)&to, &size)) <0)
{
      fprintf(stderr, "nu pot accepta conexiunea \n");
      close(sx);
      exit(errno);
```

```
}
close(sx);
printf("conectat\n");
while(1)
{
for (i=0;i<=8;i++)
{
      strcpy(nivel.canal, nr_canal);
      strcpy(nivel.analog_input, "4");
      strcpy(nivel.analog_output, "0");
      strcpy(nivel.digital_input, "0");
      strcpy(nivel.digital_output, "0");
      nr_canal[0]++;
      if(send(s,(char *) &nivel,sizeof(nivel),0)==-1)
                {
                fprintf(stderr,"can't send message\n");
                close(s);
                exit(errno);
                }
}
}                                              /* end:
while */
}
```

Baza de date de timp real

Baza de date de timp real este gândită cu o arhitectură distribuită, fiind rezidentă atât în Controllerele de Zona conectate la proces, cât şi în staţia de urmărire şi control (supervizor). Cuvânt cheie pentru relaţia între bazele de date: Număr de Ordine - reprezintă adresa de accesare a informaţiei în baza de date distribuită şi este formată din: număr controller, număr mărime în controller. Prin intermediul numerelor de ordine pot fi accesate direct informaţiile de proces aferente unei anume mărimi.

Funcţii asigurate global şi pe niveluri

Funcţii Globale

- funcţionarea în regim automat / manual;
- urmărirea parametrilor tehnologici;
- urmărirea consumurilor specifice;
- urmărirea şi stocarea informaţiei de producţie;
- urmărirea şi înregistrarea alarmelor/avariilor;
- controlul şi comanda utilajelor;
- stocarea informaţiilor pentru mentenantă;
- afişare în format grafic pe monitor color;

- interfaţă prietenoasă cu utilizatorul (meniuri, butoane).

Funcţii standard
- Control - dispaly – înregistrări;
  - structuri meniu;
  - afişare dinamică a variabilelor de proces;
  - înregistrarea activităţilor de conducere a procesului;
  - monitorizarea intrărilor şi ieşirilor proceselor;
  - listinguri şi tabele pentru:
    - utilaje (porniri/ opriri, timp de funcţionare);
    - consumuri şi producţie (medii orare, pe tură, pe zi);
    - parametri pentru funcţionarea optimă **a proceselor**.

Funcţii Administrative
- materii prime: rezerve, consum actual, consum calculat, stoc;
- producţie: cantităţi produse şi stoc pe tipuri de produse.

Funcţii orientate pe proces
- program automat de urmărire şi control pentru procesele specifice realizat şi implementat de la consola de configurare;
- posibilitatea extinderii sistemului de control prin conectarea de Controlere de Zonă configurabile la magistrala de proces.

Funcţii asigurate de nivelul de proces
- preluarea semnalelor din proces;
- prelucrare semnale preluate (conversii A/D, filtrări);
- comanda utilajelor;
- procesare alarme;
- procesare evenimente de proces, de sistem şi evenimente operator;
- urmărire şi stocare date;
- comunicaţie între controllerele de zonă şi comunicaţia cu supervizorul pentru coordonarea fluxului tehnologic;
- preluarea unor funcţii ale supervizorului prin ataşarea serială a unei console locale;
- comutare automată în regim de avarie.

Funcţii asigurate de nivelul supervizorului
- comunicaţia cu controllerele de zonă (ZC) ale nivelului ierarhic inferior (citire date şi transmitere date şi comenzi);
- pregătire date pentru sistemul de management informaţional;
- realizarea interfeţei grafice om/maşină.

Interfaţa grafică om - maşină (MMI)

Interfaţa om-maşină a staţiei de lucru conţine informaţii de personalizare a aplicaţiei şi informaţii utile pentru conducerea operativă a procesului. În plus, oferă posibilitatea unor configurări on-line şi posibilitatea selectării de către utilizator a funcţiilor pentru conducerea neoperativă. Reprezentarea grafică facilitează comunicaţia cu utilizatorul prin: butoane 3D ...apăsat - ridicat ..., butoane ...selectat - neselectat...., switch - uri …închis - deschis …, buton selectabil - neselectabil în funcţie de situaţia tehnologică, casete de dialog …OK - Cancel…, meniuri

Funcţii MMI

Pe display sunt afişate ferestre care conţin butoane care pot fi selectate sau deselectate pentru a obţine funcţia dorită, respectiv pentru a renunţa la ea (buton de tipul ON/OFF). Butonul din fereastră este reprezentat tridimensional, ceea ce premite vizualizarea stărilor „apăsat" sau „ridicat". Butonul va fi reprezentat „ridicat" numai dacă nu există o schemă sinoptică afişată pe ecran. Textul înscris pe fiecare buton reprezintă numele funcţiei apelate la apăsarea butonului.

Starea reţelei de comunicaţie

Starea comunicaţiei se semnalizează chiar prin culoarea butonului de apelare a imaginii „Stare comunicaţie". Convenţia de semnalizare a stării comunicaţiei prin culori este următoarea:

- verde = comunicaţia funcţionează în tot sistemul
- roşu = există comunicaţie defectuasă (una sau mai multe staţii din sistem nu comunică)

O imagine în detaliu a stării comunicaţiei se poate obţine de către utilizator prin apăsarea (click) butonului „Stare Comunicaţie".

Configuraţia sistemului - starea comunicaţiei

Imaginea de detaliu obţinută la apăsarea butonului din fereastra principală este o nouă fereastră care conţine configuraţia sistemului: controllerele de zonă (ZC), supervizorul şi legăturile de comunicaţie.

Scheme sinoptice

La apelarea funcţiei se obţine pe ecran o nouă fereastră care solicită de la utilizator introducerea numărului grupei dorite. Aplicaţia conţine scheme sinoptice, realizate off-line cu ajutorul editorului de scheme. Operatorul trebuie să introducă numărul schemei sinoptice dorite şi apoi să confirme apăsând tasta OK. Dacă utilizatorul a introdus numărul grupei dorite şi a confirmat, caseta de dialog va dispărea şi pe ecran va fi desenată o nouă fereastră care va conţine imaginea schemei sinoptice selectate. Titlul sinopticului din această fereastră va fi alcătuit din denumirea schemei şi numărul acesteia. Schemele sinoptice sunt alcătuite dintr-

o imagine fixă (reprezentând un proces tehnologic, anumite instalaţii sau părţi de instalaţii monitorizate şi comandate de calculatorul de proces), completată cu mărimi analogice şi binare în punctele conectate la controlerele de zonă (ZC). Aceste mărimi sunt variabile şi sunt actualizate ciclic. Valorile analogice sunt reprezentate numeric, iar cele binare prin simboluri consacrate care descriu stările mărimilor binare din proces.

Grupe funcţionale

La apelarea funcţiei se obţine pe ecran o nouă fereastră care solicită de la utilizator introducerea numărului grupei dorite. În aplicaţie sunt definite grupe funcţionale, fiecare conţinând 8 mărimi achiziţonate. Utilizatorul trebuie să introducă numărul grupei şi apoi să confirme apăsând tasta OK.

Dacă utilizatorul a introdus numărul grupei dorite şi a confirmat, caseta de dialog va dispărea şi pe ecran va fi desenată o nouă fereastră care va conţine grupa funcţională selectată. Titlul grupei din această fereastră va fi alcătuit din denumirea grupei şi numărul acesteia.

Fereastra conţine informaţiile despre fiecare din cele 8 mărimi. Pentru fiecare punct dintr-o grupă de 8 se afişează denumirea mărimii, şi pentru analogice, unitatea de măsură, valoarea mărimii, starea, valoarea în procente şi un bargraf desenat în concordanţă cu starea mărimii, iar pentru mărimi digitale starea contactului.

Valoarile şi bargrafele analogicelor, precum şi stările digitale se împrospătează permanent.

Grupe alarme

Funcţia de semnalizare a alarmelor este activă în permanentă pe display şi este reprezentată printr-o fereastră cu butoane, fiecare buton având înscris un număr care reprezintă o grupă de alarme. Butoanele reprezentând grupele vor fi reprezentate cu culoarea corespunzătoare stării grupei, cu ordinea priorităţilor stabilită mai sus. Starea grupei se determină pe baza stărilor mărimilor din grupă. Dacă cel putin o mărime din grupă este în alarmă, starea acesteia va deveni starea grupei.

Grupa curentă de alarme

Poate fi selectată din oricare din grupele de alarme apăsând butonul cu numărul grupei dorite. Imaginea grupei curente are ca titlu numărul grupei de alarme care a fost selectată şi conţine informaţii despre alarmele din grupă. Pentru o mărime din grupă de alarme se afişează valoarea (la analogice), starea (la digitale) şi un buton pentru recunoaşterea alarmei şi lansarea meniului de acţionare în caz de avarie. Fiecare mărime din grupa curentă poate fi selectată de către user pentru a valida sau invalida semnalizarea apariţiei alarmelor generate în sistem de mărimea respectivă.

Curbe de variaţie, prezintă evoluţia unei mărimi selectate, sau evoluţia mai multor parametri grupaţi din punct de vedere tehnologic, în acelaşi sistem de coordonate pe un interval de timp dat.

Logger-ul de alarme conţine în ordine cronologică toate alarmele petrecute în limita instalaţiei tehnologice supravegheate (depăşiri de valori limită, reintrări în limite normale).

Lista evenimentelor

În lista de evenimente vor apărea în ordine cronologică toate evenimentele petrecute în limita instalaţiei tehnologice supravegheate (depăşiri de valori limită, reintrări în limite normale, schimbarea poziţiei actionărilor motoare, întrerupătoare, alte stări de defect), decizii luate de către personalul de exploatare în timpul conducerii instalaţiei (comutări automat/manual, comenzi).

Mentenanţa ; pentru asigurarea mentenanţei se listează principalele agregate şi utilaje, indicându-se timpul total de funcţioanre, precum şi numărul total de porniri/ opriri sau acţionări, împreună cu data şi ora ultimei porniri.

Rapoarte

La apelarea funcţiei de rapoarte, prin intermediul unui meniu se pot selecta rapoartele dorite pentru a fi listate din următoarele:

- valori şi stări instantanee pentru mărimile din instalaţie conectate la ZC;
- medii ale parametrilor selectaţi într-un interval de timp dat (tură, zi, lună);
- logger-ul de alarme;
- tratarea situaţiilor de avarie (analiza post-avarie);
- lista evenimentelor;
- mentenanţă.

BIBLIOGRAFIE

[Adlemo,95] Aldemo, A., et all *,Towards a Truly Flexible Manufacturing System,* Jurnal of Control Eng. Practice, vol.3, No.4,pp 545-554, April 1995

[Aftarmanesh,97] Aftarmanesh, H.; Graito, C; şi alţii, *Management of distributed information in Virtual Enterprise,* Proceedings on the 4th Conference on Concurrent Enterprising, Nottingham, 1997

[Aguiar,95] Aguiar M.W.C., Weston R.H. – *A model driven approach to enterprise integration,* Int.Journal of Computer Integrated Manufacturing, vol.8, no.3, pg.210-224, 1995

[Albus,93] Albus J.S., Barbara A.J., Nagel R.N., *Theory and practice of hierarchical control,* Proc.IEEE Computer Soc.Int.Conf Compcon Falls, National Bureau of Standards, Washington DC

[Alexander,77] Alexander,C., *A Pattern Language:Towns, buildings, Construction,* Oxford University Press, New York, 1977

[Alexander,79] Alexander,C., *The timeless of building*, Oxford University Press, New York, 1979

[Alexander, 79] Alexander, C., *The Timeless Way of Building,* New-York : Oxford University Press, 1979

[Altersys,97] Altersys White Paper, *Choosing an Operating System for PCPVirgo*, Second annual review, Canada

[Alla,97] Alla, H., Bobeanu, C., Filip, F.G., *Modeling and Simulation of Complex Production Systems using Petri Nets Based Formal Approach*, Proc.11th,European Simulation Multi-Conference,Istambul,June 1997,pp.831-38

[Banks,96] Banks, J., Carson, J.S., *Discrete-Event System Simulation,*second edition, Prentice Hall, 1996

[Barber,96] Barber M.I., Weston, R.H., *Scooping Study on Business Process Reengineering: Towards Successful IT Application*, Int. Journal of Production Research., 1997.

[BIBA] BIBA, *Analyses of Socio-Technical System, Scenario, Techniques and Tools*, Bremen, Raport , March 23-24, 1995

[Bodner,97] Bodner,D.A., Damrau, M., Zhou,C. *Virtual machine models in Electronics Assembly*, Proceedings of the 1997 Dneb International Simulation Conference and Technology Showcas, Troy, MI.

[Brandin,91] Brandin B.A., Benhabib B., Wonham W.M., *Discrete Event System supervisory control applied to the management of manufacturing workcells,*

Proc.of the 7th Int.Conf.on Computer Aided Production Engineering, Cookeville TN USA, 1991, pp.527-536

[Brandin,94] Brandin B., Charbonnier F., *The supervisory control of the automated manufacturing system of the AIP*, 1994 4th Int.Conf. on integrated manufacturing and automation technology, Rensselaer, pg.319-324, 1994

[Brandin,94] Brandin B.A., Wonham W.M., *Supervisory Control of Timed Discrete Event Systems*, IEEE Transactions on Automatic Control, Vol.39, No.2, pag.329-342, 1994

[Bradatan,93] Bradatan S.I., Stanescu A.M., Dumitrache I., *Modelling an Intelligent Supervisor for Flexible Assembly Systems*, IMACS /IFAC Second International Symposium on Mathematical and Intelligent Models in System Simulation, aprilie 1993

[Brussel, 95] van Brussel, R., *Architectural and system design issues in holonic manufacturing systems,*IFAC/IFIP/IFORS Workshop in Intelligent Manufacturing Systems, Oct.1995, Bucharest, Romania

[Burgos,95] Burgos, A, şi alţii, *The Role of Integrating platforms* 1st World Congress on Intelligent Manufacturing Processes and Systems, Puerto Rico, 1995

[Buschmann,93] Buschmann, F., *Rational Architecture for Object-Oriented Software Systems*, Journal of Object-Oriented Programming, Sept.1993

[Buckley,96] Buckley, E., *GENIAL –GEN Intelligent Access Libraries Project Description 1996*, IT and Manufacturing Partnerships, J. Browne et al. (Eds) , IOS Press

[Camarinha,95] Camarinha Matos L.M., Pinheiro-Pita H., Rabelo R., Barata J., *Towards a taxonomy of CIM activities,*Int.Journal of Computer Integrated Manufacturing, vol.8, no.3, pg.160-176, 1995

[Caramihai,98] Caramihai,S.I.;Dumitrache,I.; Stanescu, A.M.; Curaj,A., *Intelligent supervising for virtual enterprise,* Proceedins on the 1st Conference on Concurent Engineering, Sinaia- Romania , 1998

[Caramihai,97], Caramihai, S. *Teza de doctorat Supervizor inteligent pentru conducerea sistemelor flexibile de fabricatie* , Bucuresti , 1997

[Camarinha,97] Camarinha-Matos, I.M.; Lima,C;Osario,L.A.: *The PRODNET platform for production planning and management in Virtual Enterprise* Proceedings on the 4th Conference on Concurrent Enterprising, Nottingham, 1997

[Cieslak,91] Cieslak R., Desclaux C., Fawaz A.S., Varaya P., *Supervisory control of Discrete Event Processes with Partial Observations*, IEEE Trans. on Automatic Control, vol.AC-33, no.3, pg.249-260, 1991

[Chen,91] Chen E., Lafortune S., *Dealing with blocking in supervisory control of DES,* IEEE Trans. on Automatic Control, 1991

[CIM-OSA,89] *CIM-OSA reference architecture specification*, ESPRIT Project no.688, ESPRIT Consortium, AMICE, Brussels, 1989

[CIM,89] *Computer Integrated Manufacturing, The CIM Enterprise,* IBM Corporation, White Plains, NY, 1989

[Coplien,95], Coplien,O.J., Schmidt, D.C., *Pattern Languages of Program Design*, Addison-Wesley Publishing Company, 1995, pp. 441-452

[Curaj,98]Curaj, A.; Stanescu A.M.; Dumitrache, I; Caramihai, S.I., *FABRICATOR – a virtual factory based concept for extended enterprise including SMEs* Proceedins on the 1st Conference on Concurent Engineering, Sinaia- Romania , 1998

[Curaj,98] Curaj, A., *Teza de doctorat –Conducerea Sistemelor de Fabricatie Integrate in Arhitecturi de Intreprindere Virtuala*, Universitatea Politehnica Bucuresti, 1998

[Dardenne,91] Dardenne, A., S.Fickas, A van Lamsweerde, *Goal-directeg Concept Acquisition in requirements Elicitation,* Proc.6th International Workshoop on Software Specification and Design, Como, Italy, 14-21

[David,92] David R., Alla H., *Du Grafcet aux reseaux de Petri,*Edition Hermes, Paris, 1992

[Dobrescu,98]Dobrescu,R.; Curaj,A., *Informational ontologies in virtual enterprise reconfiguration*, Proceedins on the 1st Conference on Concurent Engineering, Sinaia- Romania , 1998

[Dumitarche,95]Dumitrache I., Caramihai S.I., Stanescu A.M., *Intelligent supervisory for manufacturing systems*, Preprints of the 3th IFAC-IFIP workshop on intelligent manufacturing, 24-26 octombrie 1995

[Dumitrache,95]Dumitrache, I., *Genetic Algorithmes for Intelligent Control Systems*, PUB. Publ.House, Bucharest (1995)

[Dumitrache,98]Dumitrache, I.; Stanescu, A.M.; Caramihai,S.I.; Curaj, A., *Virtual Enterprise- a strategical approach to improve socio-economic stability* proceedins on the 5th Swiss, Sinaia-Romania, 1998

[Edelweiss, 95]Edelweiss, N., J.Palazzo, M.de Oliveira, *Multi-modeling of an Industrial Application with DFD, E-R and Object-Oriented Methods*, Instituto de Informatica -Universidade Federal do Rio Grande do Sul-Brazilia, 1995

[Enescu,95]Enescu, F.; Constantinescu, S;Stanescu, A.M.; *μ-CIM An implementation model for distributed shop-floor control in computer integrated manufacturing*, MSC disertation, PUB, 1995

[Fall,95] Jim Fall, *Solving the Mystery of the Black Box*, QNX News, 4th Quarter 1995, vol.9, no.4

[FitzGerald,87] FitzGerald, J; FitzGerald*, A.Fundamentals of System Analysis*, John Wiley&Sons, 1987

[Filip,91] Filip, F.G., *System Analyses and Expert System Techniques for Operative Decision Making* in System Analyses, Modeling and Simulation, nr.8, pp 395-403

[Filip,95] Filip, F.G., *Towards more Humanised Real Time Decision Support System Balanced Automation*, Systems Architecture and Design Method, pp.230-237, Chapman&Hall

[Filip,96] Filip, F.G.; Neagu, G.; Danciulescu, D., *DSSfM from Technology Transfer to Culture Transfer in Decision Making*, Proceedings on the 13th IFAC World Congres, pp.367-372, San Francisco, 1996
[Filip,98] Filip, F.G.; Neagu, G.; Danciulescu, D., *DSS in Production Control : Change Support in Decision Making*, pp.56-65, Proceedins on the 1st Conference on Concurent Engineering, Sinaia- Romania , 1998
[Fishman,87] Fishman G.S., *Principles of Discrete Event Simulation*, Wiley, 1987
[Gamma, 93] Gamma, E., Helm,R. *Design patterns: Abstraction and Reuse of Object-Oriented Design*, ECOOP 7, Proceedings. Berlin: Springler Verlag, 1993, pp. 406-431
[Gamma,93] Gamma, E., Helm, R., *Design patterns: Elements of Reusable Object-Oriented Software* , Readind, MA: Addison Wesley, 1995
[Garlan,93] Garlan, D., *An Introduction to Software Architecture. Advanced in Software Engineering and Knowledge Endineering*, Vol.I, World Scientific Publishing Company, 1993
[Gawlick,94] Gawlick, D., Obermarck, R., *Strategic Issues in Workflow Systems*, 1994, Digital Equipment Corporation, Palo Alto, California, USA
[Gibson,92] Gibson,J.E., *Taylorism and professional education In Manufacturing Systems*. Foundations of World Class Practice (Heim J.A. and Compton W.D.,eds.,)pp.149-157, Washington DC: National Academy Press
[Giua,92] Giua, A., *Petri nets as Discrete Event model for Supervisory Control*, Teza de doctorat, Rensselaer Polytechnic Institute, Troy, New York, 1992
[Goldman,94] Goldman, S.L., Nagel, R.N., Preiss, K., *Agile Competitirs and Virtual Organizations. Strategies for Enriching the Customer* N.Y. et.al.: Van Nastrand Reinhold, 1995
[Grady,97] O'Grady, P., Wen-Yau Liang, *Remote collaborative design and manufacture with modules,* Proceedins on the 4th Conference on Concurrent Enterprise, Nottingham, 1997
[Grotzinger,92] Grotzinger, S. *Feeder Assignment Models for Concurrent Placement Machines*, IIE Transactions, Vol.24, No.4, pp.31-46, 1994
[Hannam,96] Hannam, R., *Computer Integrated Manufacturing: from concepts to realisation*, UIMIST, Manchester, Addison-Wesley, 1996
[Heim,94] Heim, Joseph A., *Integrating distributed models: the architecture of ENVISION*, Computer Integrated Manufacturing. , vol.7, no.1, 47-60, 1994
[Hopp,95] Hopp, W.J., Spearman ,M.L., Factory Physics: *The Foundation of Manufacturinf Management*, Irwin Inc. Homewood, 1995
[Inan,88] Inan K.; Varaiya P., *Finitely recursive process models for discrete event systems* in IEEE Transactions on Autonomous Control, AC-33(7), pp.626-639, 1988
[ISOTC,93] ISOTC 184SC5WG1- *Framework for Enterprise Model*, ISO 1993
[Jenson,91] Jenson K., Rosenberg G. (editori) – *High-level Petri Nets:theory and Application* – Springer Verlag, 1991

[Jensen,95] Jensen, K., *Colored Petri Nets, basic Concepts, Analyses Methods and Practical Use*, vol.I+II, Springler Verlag, 1995
[Jennings,94] Jennings, N. *Cooperation in Industrial Multi-agent Systems*, World Scientific Series in Computer Science, Vol.43, 1994
[Jovane,94] Jovane,F., *The impact of new manufacturing paradigms on society and working conditions,* in Proceedings on the Conference Integration in Manufacturing, Bruxelles, pp.35-48,1994
[Kanet,88] Kanet, J.J., *MRP96: Time to Rethink Manufacturing Logistics* Production and Inventory Management 29, pp. 57-61, 1988
[Kosanke,91] Kosanke, K., *CIM-OSA: Its Role in Manufacturing Control*, ESPRIT Consortium AMICE, Bruxelles, Belgium, 1991.
[Kidd,88] Kidd P.T, *Human and computer aided manufacturing: the end of Taylorism?* In Ergonomics of Hybrid Automated Systems I (Karwowski, W, Parsaei H.R., and Wilhelm M.R., eds.), pp.145-152, Amsterdam: Elsevier, 1988
[Kidd,94] Kidd, P.T., *Agile Manufacturing,* Addison-Wesley Publishing Company, 1994
[Lin,93] Lin, G.Y., *Distributed Production Control for Intelligent Manufacturing Systems,* Ph.D. Thesis, Purdue Univ., West Laffayette , May 1993
[Levis,90] Levis, A.H., Slusky, S.L., *Migration of Control in Decision Making Organizations*, Inform.Decision Technology , No.15, 1990
[Matra] Matra Data Vision CAS - CADE documentation: *User's Guide, Primers, References Manuals*
[Menga,93] Menga, G. G++: *An Environnment for Object Oriented Design and Prototyping of Manufacturing Systems,* in W.Guver si G. Boudreaux, eds., Intelligent Manufacturing: Programming Environments for CIM New York: Springler-Verlag, 1993
[Menga,93] Menga, G., Elia, G., *Object Oriented Design of Flexible Manufacturing* Systems, Advanced Researches in Manufacturing, ch.12, pp.315-341,Mc-Graw Hill,New York 1993
[Mertens,95]Mertens, P., at.all., *Virtelle Unternehmen- ein Organisationsstruktur fur die Zukunft?* In : technologie & management, 44 Jg.,Heft2, pp.61-68,1995
[Minzberg,87] Mintzberg, H, *The strategy concept I: five Ps for strategy*,California Management Review, Fall, 11-24., 1987
[Mitller,95] Mitller, M., Schoming, A.K., *Reducing the Variance of Cycle Times in Semiconductor Manufacturing Systems* Proceedings for the International Conference on Improving Manufacturing Performance in a Distributed Enterprise: Advance Systems and Tools, 1995
[Neagu,96] Neagu. G; Filip,F.; Stanescu, A.M s.a, *Investigare paradigme de modelare a intreprinderilor* raport la contract 2064GR/1996, MCT
[Neches,91] Neches, R et al., *Enabling technology for knowledge sharing*, AI Magazine 1991, 6(4),65-85

[OSA.CIM] ESPRIT Project No.689: *CIM OSA system architecture for CIM*, Consortium AMICE Spunger Verlag Berlin, Germany, 1988

[Oliveira,97]Alvaro D. de Oliveira; Vicente I.A.; *Log.Forecast: advanced food supply chain forecast,* Proceedings on the 4th Conference on Concurrent Enterprising, Nottingham, 1997

[Pallot,97] Pallot, M., *From concurrent engineering towards concurrent enterprising*, Proc.of the 1st Romanian Workshop on Concurrent Engineering pp 3-17, Bucharest.,1997

[Palm.,96] Palmer, J.W., *Bussiness to bussiness conectivity on the Internet*, Electronic Markets Newsletter, vol.6,No.2, pp.3-4, 1996

[Perakath,95] Perakath C.B., Christopher P. M., *Towards method for acquiring CIM ontologies*, Int. J. Com. Int. Manufacturing, vol.8, no.3,225-234, 1995

[Peterson,81] Peterson J.L., *Petri Net theory and the modelling of systems*, Prentice Hall, 1981

[Ramadge,82]Ramadge P.J., Wonham W.M., *Supervision of Discrete Event Systems*, 21st IEEE Conference on Decision and Control, Orlando, Florida, pg.1228-1229, 1982

[Ramadge,89] Ramadge P.J., Wonham W.M., *The control of discrete event systems* , Proc. IEEE 1989; PROC-77(1), pp. 81-98

[Raulefs,94] Raulefs.P, *The Virtual Factory* Proc. of 13th World Computer Congress '94, vol.2, K.Brunstein, E.Raubold (Editors) Elsevier Science B.V. (North Holland) pp.18-32, 1994

[Reisig,91] Reisig W., Bibliography of Petri Nets, editor Rozenberg G. *Advances in Petri Nets*, Springer Verlag, Lecture Notes in Computer Science, pp.317-572, 1991

[Reisig,85] Reisig W., *Petri Nets,* Springer Verlag, 1985

[Rembold,85] Rembold U., Blume C., Dillman R., *Computer Integrated Manufacturing Technology and Systems*, Marcel Dekker, Inc., NY, 1985

[Rembold,93] Rembold U., Nnaji B.O., Storr A., *Computer Integrated Manufacturing and Engineering* , Addison Wesley Publ.Company, 1993

[Reenskaug,92] Reenskaug, T., *Intermediate Smalltalk, Practical Design and Implementation*. Tutorial TOOLS Europe, 1992

[Richards,97] Richards, H.D., Dudenhausen, H-M., de Ridder, L., *Flow of orgers through a Virtual Enterprise: Their proactive planning& scheduling and reactive control,* Computing&Control Engineering Journal, 1997

[Ross,77] Ross E.T. , *Structured Analysis for Requirements Definition*, IEEE Transactions on Software Engineering, Software Engineering Vol. 3, pp. 6 - 15, 1977.

[Sauer,96] Sauer, A., Thonemann, H-G. *The Global Engineering Network and Virtual/Extended Enterprises*, IT and Manufacturing Partnerships, J.Browne et al.(Eds), IOS Press, 1996

[Sargent,93] Sargent, P., I*nherently Flexible Cell communications : a review*, Journal of CIM, vol.6, nr.4, pp.234-259, 1993.
[Scheer,89] Scheer, A.W., *Enterprise-Wide Data Modelling*, Springer-Verlag, 1989.
[Sand.,95] Sandholm, T.W., *On Automated Contacting in Multi-agent Manufacturing,* Proceeding Performance in a Distributed Enterprise : Advanced Systems and Tools, Scotland 1995
[Standard, 93] *Draft Federal Information Processing Standards Publication* 183, National Institute of Standards and Technology, USA, 1993
[Stanescu,95] Stanescu A.M., Curaj A., Patras R., *Hierarchical Object-Oriented Mission Planning for Multilegged Autonomous Robot* , Proc.IFAC/IMS, Bucuresti 1995
[Stanescu,96] Stanescu A.M., Curaj A., s.a., *Modified timed Petri Net modelling for Manufacturing Systems Performance Evaluation* CONTI'96 , Timisoara
[Stanescu,97] Stanescu A.M., Dumitrache I., Curaj A., *Hybrid Framework for Concurrent Engineering,* First Romanian Workshop in Concurrent Engineering, Bucuresti 1997
[Stanescu,97] Stanescu A.M., Dumitrache I., Curaj A., Catana D., *A new Approach for Flexible Manufacturing Control System Design* , Proceeding on the CSCS 11, Bucuresti 1997
[Stanescu,97] Stanescu, A.M., Dumitrache,I., Curaj, A. *Multi-mode based system design within virtual factory synergetic framework,* Proceedings of the 4th International Conference on Concurrent Enterprising, ICE'97, The University of Nottingham, UK, 1997
[Stanescu,96] Stanescu A.M., Caramihai S.I.,Curaj A.; Catana D., Popescu O., Popescu C., *Sisteme Dinamice cu Evenimente Discrete*, Tipografia UPB, 1996
[Stanescu,97] Stanescu A.M., *New Approach for Advanced Control in Flexible Manufacturing Systems* Proc. the 11th Control Systemsand Computer Science Conf. , Buch., May 1997
[Tard.,95] Tardif, V., *Detecting and Corecting Scheduluing Infesiabilities in a Multi-level, Finite Capacity production Environment*, Doctoral disertation , Departament of Industrial Engineering and Management Science, Northwestern University, 1995
[Tuikka,97] Tuikka T., *Towards basic virtual prototyping support for concurent design*, Proceedings of the 4th International Conference on Concurrent Enterprising, ICE'97, The University of Nottingham, UK, 1997
[Vernadat,93] Vernadat, F., *Tutorial : Enterprise Modeling and CIM, 1993*
[Ziegler,90] Ziegler B.P., Object-Oriented Simulation with Hierarchical Modular Modes, Intelligent Agents and Endomorphic Systems Academic Press, San Diego, 1990.
[Zhou,90] Zhou M.C., *A theory for the synthesis and augmentation of Petri nets in Automation*, Teza de doctorat, Rensselaer Polytechnic Institute, Troy NY, 1990

[Zhou,91] Zhou M.C., DiCesare F., *Parallel and Sequential Mutual exclusions for Petri net modelling of Manufacturing Systems with Shared Resources* – IEEE transactions on robotics and automation, vol.7, no.4, pg.515-527, 1991
[Zhou,93] Zhou M.C., DiCesare F., *Petri Net Synthsis for Discrete Event Control of Manufacturing Systems*, Kluwer Ac.Publisher, 1993
[Warnecke,93] Warnecke H-J, *The Fractal Company- A revolution in Corporate Culture*, Springer-Verlag, 1993
[Wong,96] Wong K.C., Wonham W.M., *Hierarchical control of timed discrete event systems,* Discrete Event Dynamic Systems, no.6, Kluwer Academic Pubilishers, pp.275-306, 1996
[Wilhelm,92] Wilhelm,W.E., Research Directions for Electronics Manufacturing, IIE Transactions, Vol.24,No.4, pp.6-17, 1992
[Westkamper,97] Westkamper, E., Dudenhausen, H.M, *Optimization of Order flow through virtual enterprises Facilitating Deployment of Information and Communication technology for Competitive Manufacturing* Ed.D.Fichter, Proceedings on IiM, 1998

Dr. Ing. Adrian Curaj (n. 1958), absolvent al Facultăţii de Electronică şi Telecomunicaţii, Institutul Politehnic Bucurest, in anul 1983. Sef.lucrari la Facultatea de Automatică şi Calculatoare a Universităţii POLITEHNICA din Bucureşti.
Autor de cursuri universitare de specialitate şi a numeroase lucrări ştiinţifice publicate în ţară şi în străinătate.
Membru al Societăţii Române de Robotică şi al Societăţii Române de Inginerie Concurentă.

ISBN: 973-30-9925-2