

**CONTRIBUȚII LA DEZVOLTAREA UNOR STRUCTURI
DESCRISE DE AUTOMATIZARE A FABRICAȚIEI
PENTRU HIPERSISTEME ROBOTIZATE DE TIP CIM**

Teză de doctorat

Conducător științific

Prof.dr.Ing.Dr.h.c. Kovács Francisc Viliam

Doctorand

Șef.lucr.Ing. Barabás Tiberiu

2003

CUPRINS

1. TEMA TEZEI	7
1.1. Motivația tezei.....	7
1.2. Conținutul tezei.....	8
2. INTRODUCERE ÎN STRUCTURILE DE AUTOMATIZARE A FABRICAȚIEI UTILIZATE ÎN CADRUL HIPERSISTEMELOR CIM	11
2.1. Noțiuni introductive	11
2.2. Structura de comandă DNC a mașinilor unelte.....	12
2.3. Structuri tipice de automatizare a fabricației la nivelul celulelor de fabricație ...	14
2.3.1. Structură de comandă cu conectare directă la controlerul de celulă	15
2.3.2. Structură de comandă cu interconectare printr-o rețea locală LAN.....	16
2.4. Structuri tipice de automatizare a fabricației la nivelul compartimentelor de execuție al unui Hipersistem CIM	22
3. ANALIZA EVOLUȚIEI STRUCTURILOR DESCHISE DE COMANDĂ DIN CADRUL HIPERSISTEMELOR CIM (“OPEN CONTROL”)	31
3.1. Introducere în comanda de la distanță a echipamentelor de fabricație.....	31
3.1.1. Ce este comanda de la distanță a unui echipament de fabricație integrat într-un Hipersistem CIM?	31
3.1.2. Modul de realizare a comenzii de la distanță	33
3.2. Cauzele inițiale care au condus la introducerea în Hipersisteme CIM a structurilor deschise de comandă	35
3.3. Elaborarea și introducerea primelor structuri deschise de comandă în Hipersisteme CIM	37
3.3.1. Care sunt problemele principale ale transmiterii comenzilor în Hipersisteme CIM?.....	37
3.3.2. Elaborarea unui model standard de transmitere a comenzilor între două structuri deschise - Modelul de referință OSI	39
3.3.3. Elaborarea unor metode standarde de acces la mediul de transmitere a comenzilor	45
3.3.3.1. Metoda de acces la magistrala cu jeton (Token Bus).....	45
3.3.3.2. Metoda de acces cu sesizarea purtătoarei (Ethernet)	48
3.3.4. Elaborarea unor metode standarde de transmitere referitoare la mediul fizic	49
3.3.5. Apariția primelor structuri deschise destinate transmiterii comenzilor în Hipersisteme CIM. Apariția MAP-ului	51
3.3.5.1. Arhitectura Full-MAP	52
3.3.5.2. Arhitectura Mini-MAP.....	55
3.3.5.3. Succesele inițiale ale MAP-ului	57
3.3.6. Standardizarea comenzilor utilizate în cadrul structurilor de fabricație automatizată. Apariția MMS-ului	61
3.3.6.1. MMS - primul standard deschis de comandă de la distanță	61
3.3.6.2. Structura funcțională a MMS-ului. Echipamentul Virtual de Fabricație.....	63
3.3.6.3. Încercări privind elaborarea unor structuri MMS specifice pentru fiecare tip de echipament de fabricație.....	68
3.3.7. Necesitatea elaborării unor variante cu cost redus ale structurilor deschise de comandă utilizate în Hipersisteme CIM.....	69

3.4. Introducerea în Hipersisteme CIM a unor structuri deschise de comandă cu cost redus	69
3.4.1. Introducerea unor structuri deschise ale "internetului" în Hipersisteme CIM	70
3.4.1.1. Implementarea MMS-ului sub TCP/IP. Specificația firmei Boeing	70
3.4.1.2. Proiectele "Wave I - III" ale firmei General Motors	72
3.4.1.3. Specificația FL-net a grupului JOP din Japonia.....	75
3.4.2. Introducerea unor structuri deschise de comandă destinate inițial domeniului proceselor industriale.....	75
3.4.2.1. Structuri deschise de comandă cu magistrale de câmp	75
3.4.2.2. Structuri deschise de comandă cu protocol "CIP"	79
3.5. Concluzii	83
4. ANALIZA TENDINTELOR DE DEZVOLTARE PENTRU HIPERSISTEME CIM A UNOR ECHIPAMENTE DE COMANDĂ CU ARHITECTURĂ DESCHISĂ ("OPEN CONTROLLER")	85
4.1. Necesitatea introducerii în Hipersisteme CIM a echipamentelor de comandă cu arhitectură deschisă.....	85
4.2. Dezvoltarea echipamentelor de comandă robot (ROC) cu arhitectură deschisă	87
4.2.1. Evoluția echipamentelor de comandă robot către o arhitectură deschisă.....	88
4.2.2. Exemple de echipamente de comandă robot cu arhitectură deschisă ...	93
4.2.2.1. Controler Bosch IQ200 al robotului Turboscara SR6/SR7.....	93
4.2.2.2. Controler KR C1 al unui robot KUKA.....	95
4.3. Dezvoltarea echipamentelor de comandă numerică (CNC) cu arhitectură deschisă	97
4.3.1. Clasificarea echipamentelor CNC cu arhitectură deschisă	97
4.3.2. Proiecte majore de cercetare legate de dezvoltarea unor echipamente CNC cu arhitectură deschisă.....	100
4.3.2.1. Proiectul OSACA (Open System Architecture for Controls within Automation systems).....	100
4.3.2.2. Proiectul OMAC (Open Modular Architecture Controller)	103
4.3.2.3. Proiectul OSEC (Open System Environment for Controllers)....	105
4.4. Concluzii	106
5. ELABORAREA UNOR MODELE DESTINATE STRUCTURILOR DESCHISE DE COMANDĂ A ROBOȚILOR INDUSTRIALI.....	107
5.1. Introducere	107
5.2. Elaborarea unui model al resurselor fizice ale unui robot industrial.....	108
5.2.1. Resursele fizice ale subsistemului manipulator.....	110
5.2.2. Resursele fizice ale subsistemului de conducere.....	112
5.2.3. Echipamente auxiliare.....	116
5.2.4. Programul de aplicație robot	116
5.2.5. Sisteme de coordonate robot utilizate la comanda de la distanță a subsistemului manipulator.....	117
5.3. Elaborarea unui model al resurselor logice ale unui robot industrial.....	118
5.3.1. Capacitatea de funcționare	118
5.3.2. Tipul controlului suportat.....	119

5.3.3. Starea de sincronizare	119
5.3.4. Starea de funcționare	120
5.3.5. Starea de alimentare cu energie	123
5.4. Elaborarea unui model de accesare de la distanță a resurselor fizice și logice ale unui robot industrial	123
5.4.1. Prezentarea modelului	125
5.4.2. Stabilirea atributelor modulelor definite	127
5.5. Concluzii	133
6. ELABORAREA UNOR STRUCTURI DESCHISE DE COMANDĂ PENTRU UN MINI-SISTEM CIM	135
6.1. Introducere	135
6.2. Prezentarea mini-sistemului CIM2000	135
6.2.1. Structura detaliată a mini-sistemului	138
6.2.2. Arhitectura sistemului de comandă	141
6.3. Elaborarea unui proiect de introducere în mini-sistemul CIM2000 a unor structuri deschise de comandă	146
6.3.1. Prezentarea proiectului	146
6.3.2. Rezultate obținute	150
6.4. Elaborarea unor structuri de comandă de la distanță a robotului RV-M1 din cadrul mini-sistemului CIM2000	152
6.4.1. Conectarea informațională a controlerului de robot RV-M1 la un calculator PC	152
6.4.2. Elaborarea unor structuri de comandă pentru accesarea de la distanță a resurselor fizice și logice ale robotului RV-M1	156
6.4.3. Implementarea diagramei stărilor de funcționare	161
6.5. Elaborarea unor structuri deschise MMS de semaforizare a accesului la resurse comune în cadrul mini-sistemului CIM2000	164
6.6. Concluzii	166
7. EXPERIMENTAREA UNOR STRUCTURI DESCHISE DE COMANDĂ PRIN CONSTRUIREA UNEI CELULE FLEXIBILE DE FABRICAȚIE INTEGRAT CU CALCULATORUL	169
7.1. Introducere	169
7.2. Prezentarea componentelor celulei flexibile de fabricație	170
7.3. Conceperea și realizarea sistemului de comandă	174
7.3.1. Conectarea informațională a echipamentului CNC al centrului de prelucrare CP20UO la un calculator PC	177
7.3.2. Conectarea informațională a microcalculatorului de comandă al depozitului MR800 la un calculator PC	182
7.3.3. Conectarea informațională a controlerului robotului AID la un calculator PC	186
7.3.4. Conectarea informațională a calculatoarelor PC de interfață la controlerul de celulă	188
7.4. Concluzii	195
8. CONTRIBUȚII ORIGINALE. CONCLUZII FINALE	197
BIBLIOGRAFIE	199

CAPITOLUL 1.

TEMA TEZEI

1.1. Motivația tezei

Odată cu apariția Hipersistemelor CIM a apărut și problema costurilor generate de integrarea în acestea a echipamentelor de fabricație. O parte din aceste costuri sunt cauzate de integrarea informațională, a echipamentelor de comandă cu care sunt dotate, în structurile de automatizare a fabricației din cadrul Hipersistemelor CIM.

Pentru reducerea costurilor de integrare informațională, pe plan mondial au fost promovate o serie de proiecte care în esență urmăresc dezvoltarea și introducerea în Hipersisteme CIM a unor **structuri deschise de automatizare a fabricației**. Aceste structuri sunt caracterizate prin faptul că au la bază specificații deschise, adică accesibile atât producătorilor potențiali cât și utilizatorilor, prin care sunt descrise modalitățile concrete de realizare a comenzii în cadrul lor.

Trebuie arătat că în ultimii 10-15 ani existau deja mai multe încercări (cu succese mai mici sau mai mari), pentru rezolvarea problemei de dezvoltare, a unor astfel de structuri "deschise", pentru Hipersisteme CIM. Problema a intrat din nou în actualitate după ce **conceptul sistemelor deschise** ("Open System") a devenit o cerință hotărâtoare în domeniul sistemelor de calcul (îndeosebi în tehnologia PC) respectiv în sisteme distribuite pe bază de "Internet".

Open Architecture Control for the Robotics Industry	Orlando, FL	9-10 februarie 2000
Near-Term Objectives Workshop – Open Architecture Control for Robotics	Ypsilanti, Michigan	28 iunie 2000
Near-Term Objectives Workshop – Open Architecture Control in Robotics	Rochester Hills, MI	29 ianuarie 2001
4th RIA/NIST Workshop on Open Architecture Control in Robotics	Chicago, IL	8 iunie 2001
5th RIA/NIST Workshop on Open Architecture Control in Robotics	FANUC Robotics North America	9 ianuarie 2002
6th RIA/NIST Workshop on Open Architecture Control in Robotics	Orion, MI	4 iunie 2002

Tab.1.1. Workshop-uri pe tema structurilor deschise de automatizare a fabricației

Astfel, în ultimii 2-3 ani au fost organizate o serie de simpozioane și "mese de lucru" (workshop) legate de această temă, tab.1.1. În cadrul acestora au fost promovate:

1. Dezvoltarea unor structuri de comandă deschisă (de la distanță), a echipamentelor de fabricație din cadrul Hipersistemelor CIM ("**Open Control**"),
2. Dezvoltarea unor echipamente de comandă în variantă cu arhitectură deschisă ("**Open Controller**").

Prezenta teză se încadrează în această direcție a cercetărilor de pe plan mondial, încercând să aducă contribuții la dezvoltarea unor structuri deschise de comandă, independente de echipament respectiv de furnizor, destinate Hipersistemelor CIM robotizate.

1.2. Conținutul tezei

În capitolului 2. se face o mică introducere în Hipersisteme CIM, prin prezentarea structurilor tipice de automatizare a fabricației din cadrul acestora.

În capitolele 3. și 4., în concordanță cu direcțiile de cercetare amintite în paragraful anterior, se analizează evoluția în cadrul Hipersistemelor CIM a structurilor deschise de comandă ("Open Control") respectiv tendințele de dezvoltare a unor echipamente de comandă cu arhitectură deschisă ("Open Controller").

În capitolul 3., pe lângă prezentarea primelor structuri deschise de comandă (MAP, MMS), care au la bază Modelul OSI de interconectare a "sistemelor deschise", se analizează și cauzele insucceselor parțiale ale acestora. În căutarea unor soluții "mai economice", se identifică două mari tendințe pe plan mondial: introducerea unor structuri deschise ale "Internetului" în interconectările industriale (MMS sub TCP/IP, proiectele "WAVE I - III", specificația FL-NET), respectiv introducerea unor structuri deschise, utilizate în domeniul proceselor industriale. Astfel, sunt menționate și analizate unele structuri deschise de comandă cu magistrale de câmp (Interbus, Foundation Fieldbus, Profibus), respectiv unele structuri noi de comandă (DeviceNet, ControlNet, EtherNet/IP) care au la bază un protocol comun CIP (Control and Information Protocol).

În capitolul 4., analiza pleacă de la tehnologia PC, care este considerată în rândul sistemelor de calcul comerciale, un standard "de facto". Se identifică tendințele actuale de introducere a acestei tehnologii în dezvoltarea unor arhitecturi deschise pentru echipamente de comandă robot (controlere de robot), respectiv pentru echipamente de comandă numerică (CNC), destinate mașinilor-unelte. În cazul controlerelor de roboți, tendințele prezentate sunt exemplificate prin analiza a două arhitecturi "deschise": Bosch IQ200 al robotului Turboscara SR6/SR7 respectiv KR C1 al robotului KUKA. În cazul echipamentelor CNC se identifică 3 mari categorii, în funcție de "gradul de deschidere" al arhitecturii de comandă. De asemenea se analizează principalele proiecte de cercetare de pe plan mondial (OSACA, OMAC, OSEC-JOP), care au ca scop dezvoltarea unor arhitecturi de comandă "complet deschise" respectiv independente de producător.

În capitolele 5., 6. și 7. sunt prezentate contribuțiile principale la dezvoltarea unor structuri deschise de comandă, aferente automatizării fabricației în cadrul Hipersistemelor CIM.

În capitolul 5., se prezintă contribuțiile la elaborarea unor modele destinate structurilor deschise de comandă a roboților industriali din cadrul Hipersistemelor CIM, și anume: *modelul resurselor fizice*; *modelul resurselor logice* respectiv *modelul accesării de la distanță a resurselor fizice și logice ale unui robot industrial*. Prin primele două modele se încearcă stabilirea și definirea acelor resurse fizice și logice ale

roboților industriali, care în general pot fi sau sunt necesare să fie accesate prin comenzi “de la distanță” (adică prin rețeaua informațională a Hipersistemelor CIM). Prin al treilea model se propune un nou mod de accesare de la distanță a resurselor fizice și logice ale unui robot industrial. Pe baza acestui model, accesul de la distanță la controlul resurselor se reduce la accesul standard la modulele software ale unui controler de robot cu arhitectură deschisă și modulară. În acest capitol, de asemenea sunt stabilite și funcțiile standarde ale principalelor module, luând în considerare cerințele generale ale realizării comenzii de la distanță a robotului industrial.

În capitolul 6., se prezintă contribuțiile la dezvoltarea unor structuri deschise de comandă pentru un mini-sistem CIM. Întâi se prezintă mini-sistemul CIM2000 de la Universitatea din Oradea respectiv structura “originală” a sistemului de comandă al acestuia. Apoi este prezentat un proiect de dezvoltare a sistemului de comandă (realizat pentru facilitarea introducerii unor structuri deschise) respectiv rezultatele “parțiale” ale implementării acestuia. Tot în acest capitol se prezintă principalele structuri “deschise” elaborate, pentru comanda de la distanță a robotului industrial RV-M1 din mini-sistemul CIM2000: *conectarea controlerului robot la un calculator PC; implementarea pentru robotul RV-M1 a modelului de accesare de la distanță a resurselor fizice și logice* elaborat în capitolul 5.; *implementarea diagramei cu stările de funcționare*. În final se prezintă elaborarea unor structuri deschise MMS, pentru semaforizarea accesului la resursele comune, din cadrul mini-sistemului CIM2000.

Trebuie arătat că dezvoltarea și apoi experimentarea unor structuri deschise de comandă, la nivelul întregului mini-sistem CIM2000 nu era posibilă, datorită unor protocoale brevetate respectiv datorită unor interfețe utilizator “închise”, a căror specificații nu au fost accesibile. În consecință era necesară construirea unei noi structuri de fabricație integrat cu calculatorul, de exemplu a unei celule flexibile de fabricație, care să fie prevăzută cu un sistem deschis de comandă.

În capitolul 7., se prezintă construirea acestei celule, prin integrarea fizică și informațională a unor module de fabricație adecvate (roboți industriali, mașini unelte cu comandă numerică, depozit tampon), existente în cadrul Universității din Oradea. Scopul a fost dezvoltarea și experimentarea unor structuri deschise de comandă, care să asigure comanda celulei printr-o serie de comenzi “standarde”, independente de echipament. În acest scop a fost conceput un model general al stărilor de funcționare, uniform valabilă pentru fiecare “stație” în parte. În cadrul capitolului se prezintă întâi componentele celulei, apoi etapele conceperii și realizării sistemului de comandă: conectarea informațională a echipamentelor de comandă la calculatoarele PC de “interfață” și conectarea acestora din urmă la “controlerul de celulă”. De asemenea se prezintă arhitectura client-server a sistemului de comandă, prin prezentarea programelor “Server” instalate pe calculatoarele PC de interfață și a programului “Client” instalat pe controlerul de celulă. Se prezintă mediul de programare aplicație, creat special pentru experimentare respectiv programul “test” de comandă a celulei de fabricație.

În capitolul 8. se prezintă sub o formă sintetică, contribuțiile originale la tema tezei precum și concluziile finale.

CAPITOLUL 2.

INTRODUCERE ÎN STRUCTURILE DE AUTOMATIZARE A FABRICAȚIEI UTILIZATE ÎN CADRUL HIPERSISTEMELOR CIM

2.1. Noțiuni introductive

Prin **fabricație** se înțelege totalitatea activităților – umane sau automate – care concurează la realizarea unui produs de o anumită concepție și o anumită destinație dată, pornindu-se de la starea de semifabricat și până la starea de produs finit, [71]. În acest context, prin **automatizarea fabricației** se înțelege “automatizarea” activităților enunțate mai sus. Ca rezultat al automatizării, aceste activități vor putea fi efectuate în mod automat, adică fără intervenția operatorului uman. O **fabricație complet automatizată** este caracterizată prin efectuarea în mod automat, a tuturor activităților legate de realizarea produsului.

Structura de bază a fabricației automatizate este **celula de fabricație**. O celulă de fabricație este destinată în general pentru fabricarea în mod automat a unor piese care fac parte dintr-o familie tipodimensională. Astfel ele se constituie pe grupe de operații tehnologice cum sunt: prelucrări prin așchiere piese prismatice, prelucrări prin așchiere piese de revoluție, deformări plastice, tratamente termice, montaj, vopsire, sudură etc.

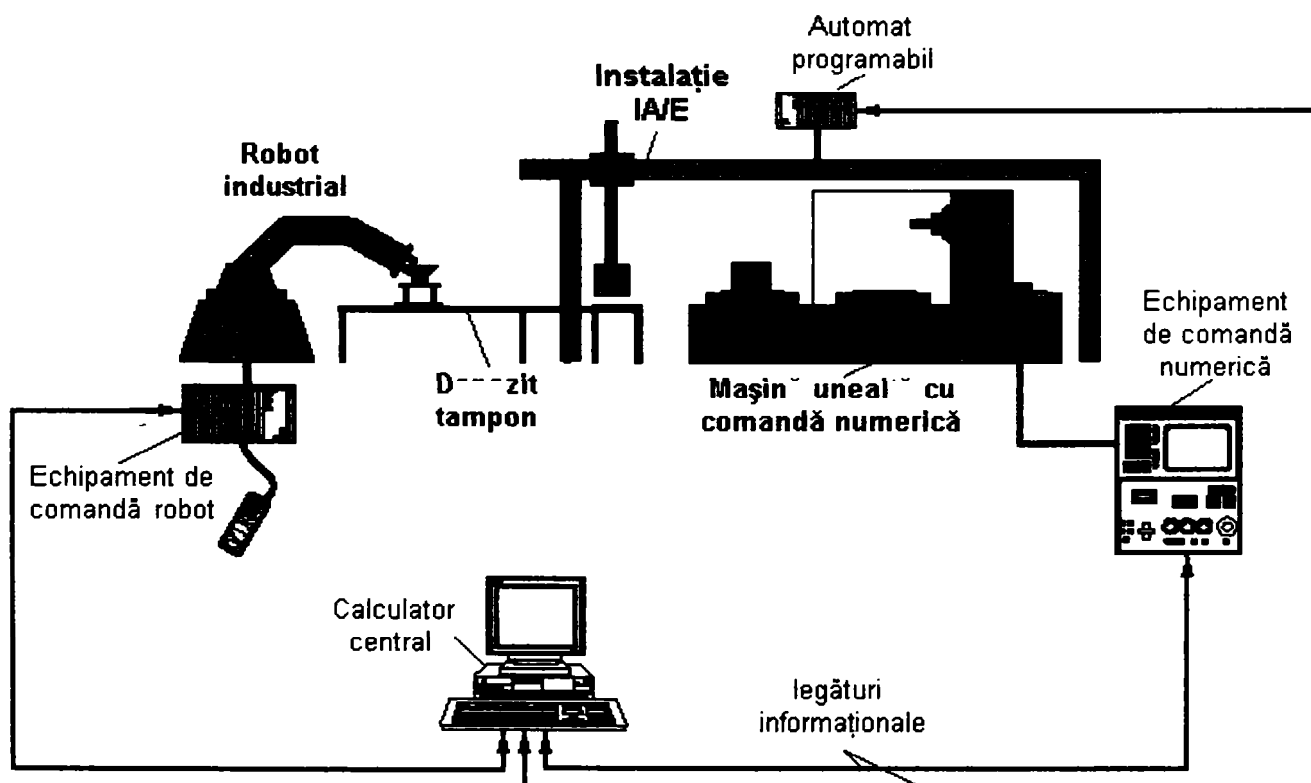


Fig.2.1. Componentele principale ale unei celule de fabricație

O **celulă de fabricație automatizată** conține unu sau mai multe mijloace de producție automatizate (de exemplu mașini unelte cu comandă numerică), depozite

tampon pentru semifabricate și piese finite respectiv unu sau mai multe sisteme de transfer automat al pieselor între depozite și mijloace de producție, fig.2.1. Aceste sisteme de transfer automat al pieselor pot fi realizate prin intermediul unor "instalații aducătoare/ de evacuare" (IA/E) - în cazul mișcărilor simple de transfer și poziționare – și/sau prin intermediul roboților industriali (în cazul în care sunt necesare mișcări complexe de manipulare), [71].

Toate activitățile de fabricație din cadrul unei celule sunt sub coordonarea automată a unui "**calculator central**". Această coordonare automată necesită existența unor legături informaționale între echipamentele de comandă din cadrul celulei și calculatorul central.

Integrarea cu "calculatorul" într-o întreprindere atât a activităților de fabricație, cât și a activităților legate de:

- de concepția (proiectarea) constructivă și tehnologică a produselor,
- de planificarea, programarea și urmărirea producției,
- de logistică, financiară, aprovizionare și de desfacere,
- respectiv de conducere (management) a întreprinderii

este cunoscută sub numele de **CIM (Computer Integrated Manufacturing)**, [71]. Astfel termenul CIM se referă la integrarea "informațională" a activităților de producție iar fabricația este de fapt doar una dintre activitățile producției, [69].

Observații:

1. Conform celor de mai sus, mai potrivită ar putea fi denumirea "Computer Integrated Production" - Producție Integrată cu Calculatorul (CIP), [69].
2. Termenul de **Hipersistem CIM**, este utilizat în general pentru indicarea integrării informaționale a "subsistemelor" de producție, într-un sistem complex și unitar.

2.2. Structura de comandă DNC a mașinilor unelte

Acest tip de structură este utilizată pentru Comanda Numerică Directă (Direct Numerical Control – DNC) a mașinilor-unelte din cadrul unei celule de fabricație. În esență, această structură de comandă, numită și **structură DNC**, oferă posibilitatea introducerii programelor de prelucrare, în memoria echipamentelor de comandă numerică a mașinilor respective în mod "direct", de pe un calculator central (pe care sunt stocate), în locul introducerii manuale sau de bandă perforată de către operatorul uman.

Între calculatorul central (denumit și **calculator DNC**) și echipamentele de comandă numerică din cadrul celulei există o legătură informațională bidirecțională (de obicei legătură serială RS232c). Astfel calculatorul DNC, pe lângă transferul către echipamentele de comandă numerică a programelor de piesă (respectiv a unor date, cum sunt corecții de scule), are acces la date (informații) referitoare la starea de funcționare a mașinilor și în consecință poate iniția comenzi pentru efectuarea anumitor operații.

În fig.2.2. s-a prezentat un exemplu de conectare informațională a echipamentelor de comandă numerică, într-o structură de comandă de tip DNC, așa cum

există la firma REMEX Corp., SUA., [97]. Pe baza figurii, legătura informațională între calculatorul DNC și echipamentul de comandă numerică NC (Numerical Control) este realizată prin intermediul unor **Interfețe de Mașină** (MIU – Machine Interface Unit).

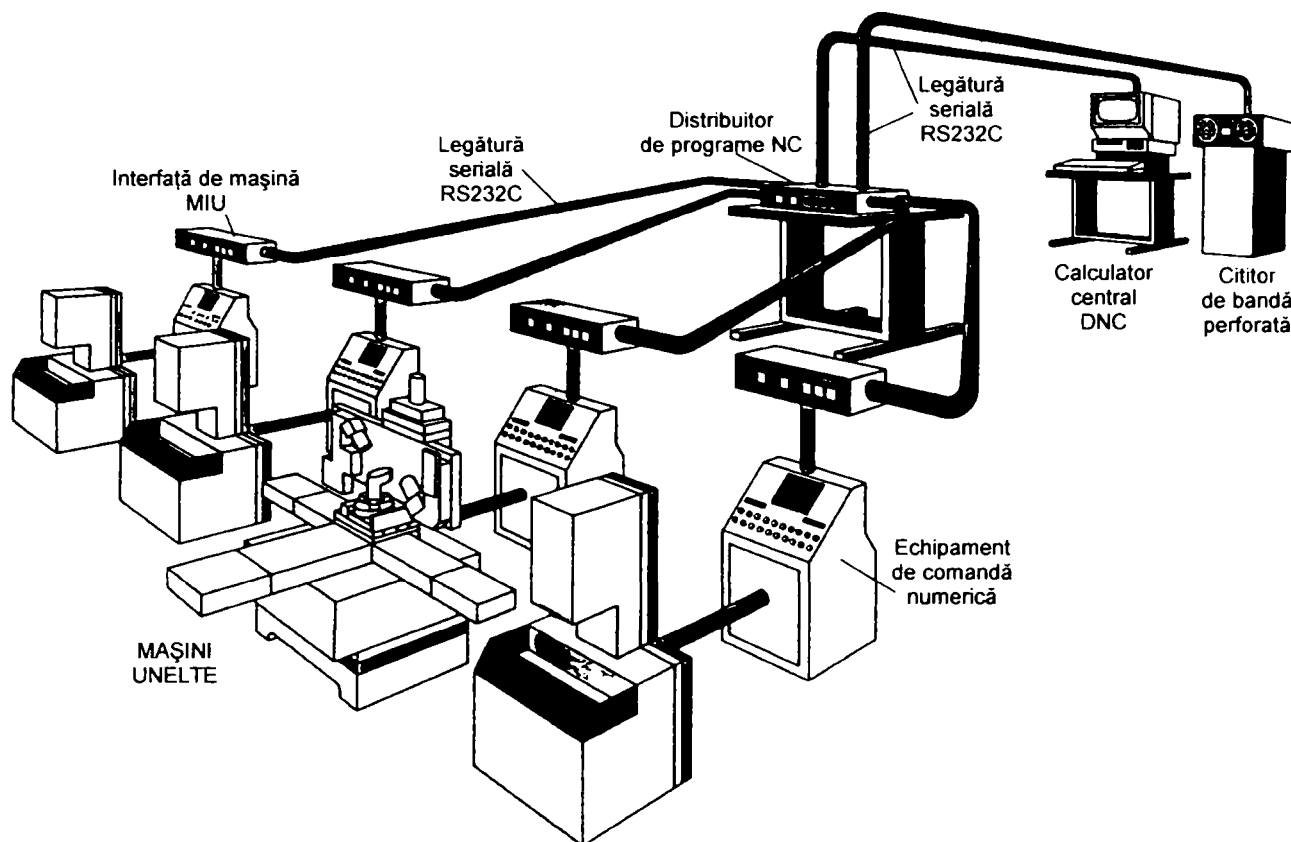


Fig.2.2. Exemplu de structură de comandă de tip DNC, [97]

În unele cazuri aceste “interfețe de comunicație” sunt instalate în interiorul echipamentelor, caz în care au denumirea de “**interfață DNC**”. În alte cazuri aceste interfețe sunt sisteme cu memorii proprii, cu ecran (display) și panou operator pentru recepționarea, stocarea și transmiterea mai departe a programelor de prelucrare (fig.2.2.) În acest ultim caz, programul poate fi transmis către interfața MIU a echipamentului NC chiar în timpul execuției programului curent de prelucrare.

Programul transmis este stocat în memoria interfeței MIU până la terminarea execuției programului curent, respectiv până la încărcarea în memoria internă a echipamentului NC.

Trebuie menționat faptul că interfețele MIU nu sunt necesare în cazul echipamentelor de comandă numerică de “tip nou”: CNC (Computer Numerical Control) care de regulă au deja încorporate **interfețe DNC**.

Observație:

- Diferența esențială între un echipament de comandă numerică de tip NC (Numerical Control) și unul de tip CNC (Computer Numerical Control), este faptul că până când la primile funcții de comandă sunt implementate exclusiv prin mijloace hardware, la echipamente CNC ele sunt implementate prin mijloace software, adică au la bază un calculator numeric (Computer), format dintr-un sistem cu microprocesoare.

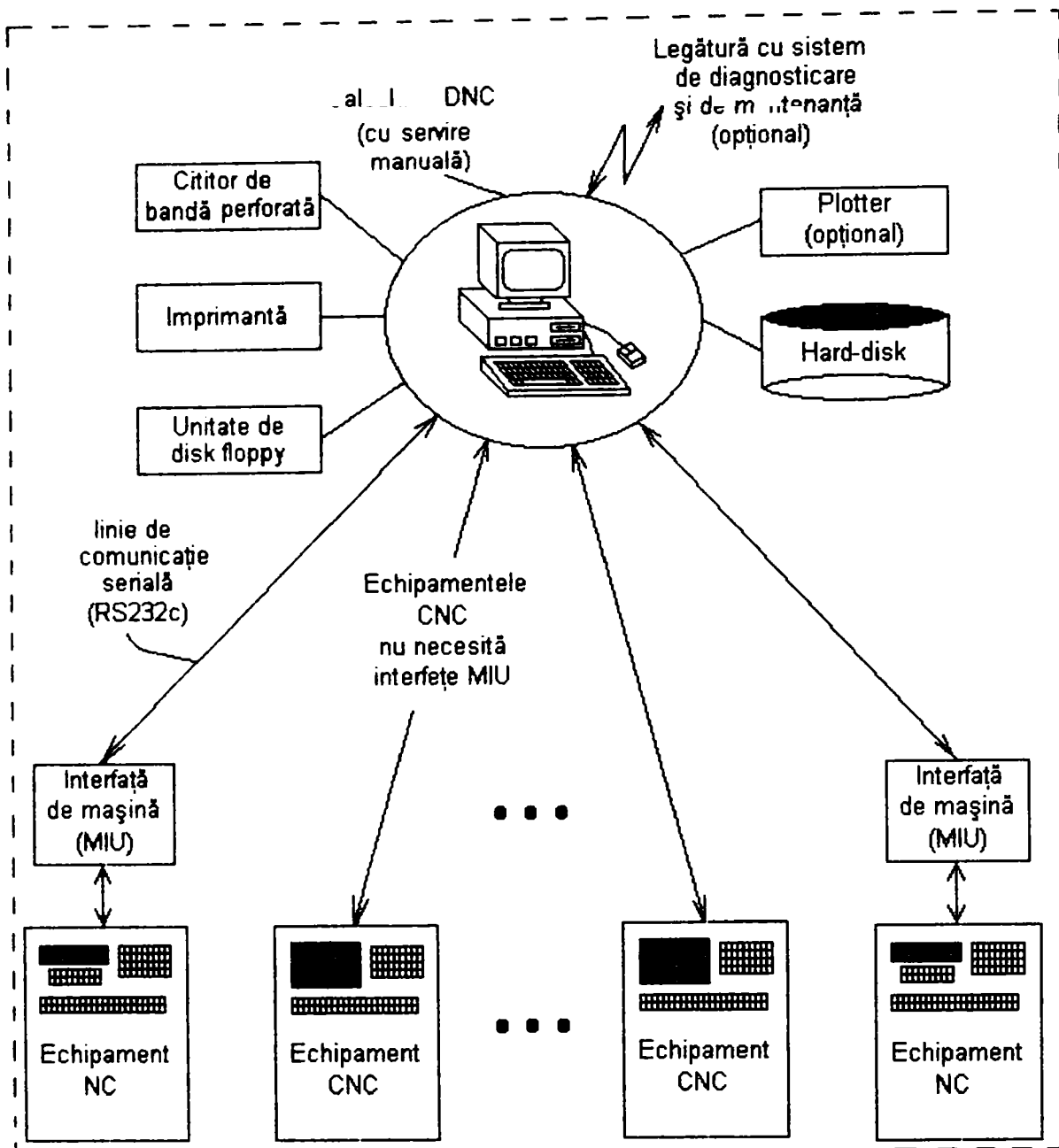


Fig.2.3. Modelul de interconectare a echipamentelor de comandă numerică într-o structură de tip DNC

Modul tipic de interconectare a echipamentelor de comandă numerică (NC, CNC), în cazul unei structuri de tip DNC, este prezentat în fig.2.3., [97]. Se poate observa că în general lipsesc legăturile informaționale cu alte structuri funcționale (proiectare, urmărirea producției) din cadrul întreprinderii. Astfel rămân modalitățile tradiționale de încărcare pe calculator a unor programe de prelucrare: de pe bandă perforată (prin utilizarea unui cititor de bandă) sau de pe unități de stocare magnetică a datelor (disk floppy). De asemenea, deservirea calculatorului DNC (introducere date și programare) se realizează prin mijloace manuale.

2.3. Structuri tipice de automatizare a fabricației la nivelul celulelor de fabricație

Pentru realizarea coordonării tuturor activităților de fabricație la nivelul unei celule este necesară conectarea la un calculator central, nu numai a echipamentelor de comandă numerică (NC și/sau CNC), dar și a echipamentelor de comandă robot

(ROC – Robot Controller) respectiv a automatelor programabile (PLC - Programmable Logic Controller). Acesta din urmă este utilizat în general pentru supravegherea proceselor discrete, legate de fluxul de piese din cadrul celulei, respectiv a instalațiilor IAE.

Calculatorul central, având **funcția de coordonare** a tuturor echipamentelor de comandă (NC, CNC, ROC, PLC) din cadrul celulei, este numit **Controler de Celulă** (CC- Cell Controller). În funcție de modul de conectare informațională a echipamentelor de comandă amintite la controlerul de celulă, distingem mai multe variante pentru structura de comandă a unei celule de fabricație:

1. Structură de comandă cu **conectare directă la controlerul de celulă**,
2. Structură de comandă cu **interconectare printr-o rețea locală LAN**.

2.3.1. Structură de comandă cu conectare directă la controlerul de celulă

Această structură de comandă derivă din structura de tip DNC, în care fiecare echipament NC și/sau CNC este conectat separat la un calculator central prin intermediul unor legături seriale RS232c, așa cum a fost prezentat în fig.2.3.

În plus față de structura DNC se conectează la acest calculator central, numit în continuare **controler de celulă**, și celelalte echipamente de comandă (ROC, PLC) din cadrul celulei de fabricație, tot prin **legături seriale** dar **distincte**, fig.2.4.

Controlerul de celulă îndeplinește **funcția de comandă/coordonare** a activităților de fabricație din cadrul celulei, prin:

- Încărcarea echipamentelor de comandă (NC, CNC, PLC, ROC) cu programe și cu date, aferente procesului curent de fabricație;
- Inițierea unor comenzi pentru intrarea în funcțiune/oprirea mașinilor de lucru, a dispozitivelor instalației aducătoare/de evacuare, a roboților respectiv a modulelor de fabricație flexibilă, [71];
- Monitorizarea stărilor de funcționare ale fiecărui echipament de fabricație;
- Gestionarea fluxului de materiale din cadrul celulei.

Analizând fig.2.4. poate fi observat ușor **caracterul ierarhic** al acestei structuri: controlerul de celulă este pe post de “calculator central” (numit și “master”) iar echipamentele de comandă conectate la el sunt pe post de “slave” (subordonat). Aceasta înseamnă că deși în general echipamentele de comandă pot deține inițiativa comunicației, în situații de concurență controlerul de celulă are prioritate.

Dezavantajele acestei structuri de comandă sunt legate de caracterul centralizat al comunicațiilor. Nici un echipament nu poate să comunice cu altul, numai prin intermediul controlerului de celulă. Acesta din urmă însă trebuie să coordoneze singur toate activitățile de fabricație din cadrul celulei, ceea ce este o sarcină extrem de mare. Datorită vitezelor (relativ) mici de transfer, utilizate la transmisia serială RS232c, coordonarea în timp real a activităților de fabricație, de multe ori este imposibilă.

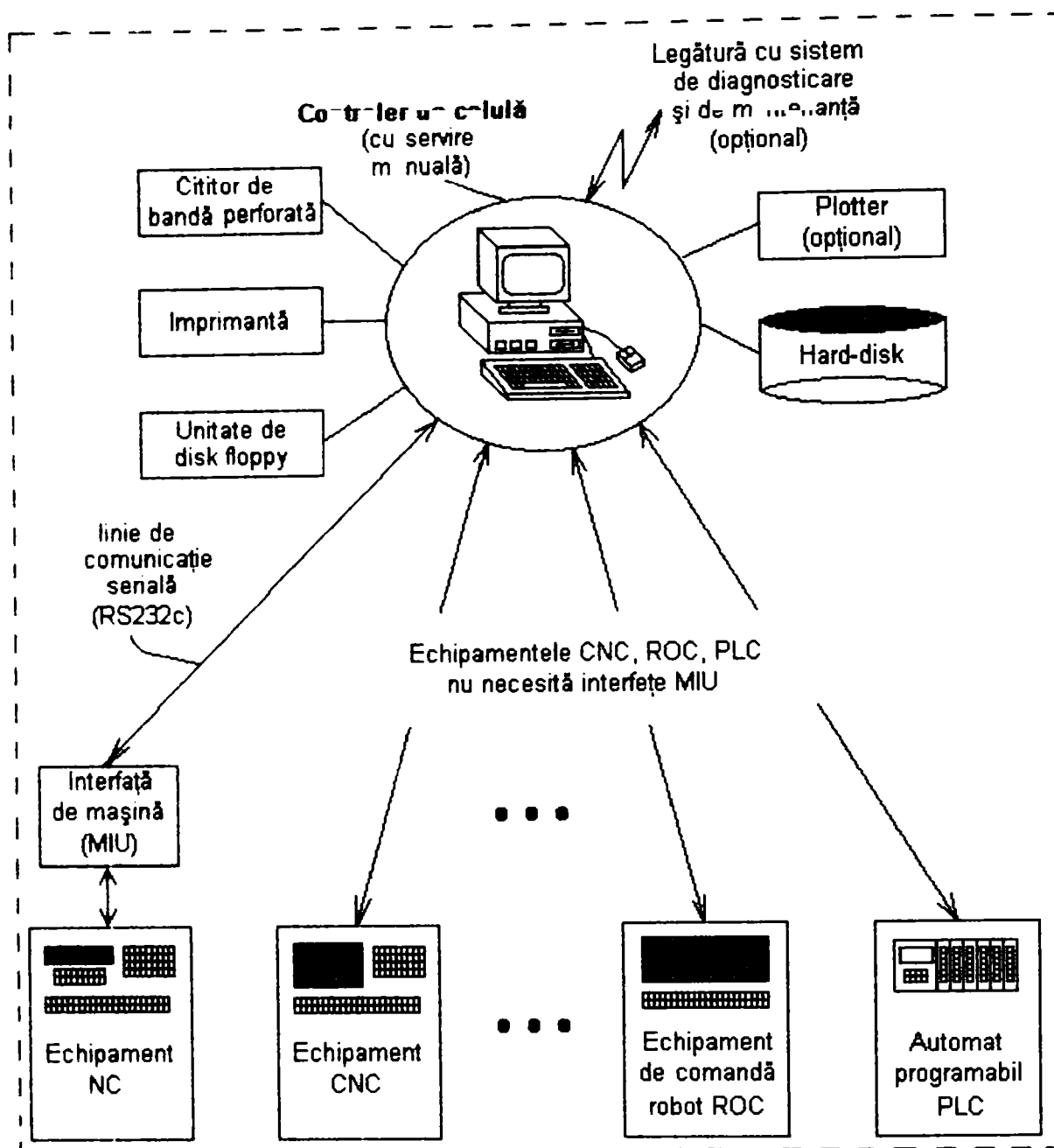


Fig.2.4. Modelul de conectare directă a echipamentelor de comandă din cadrul unei celule de fabricație la controlerul de celulă

Comanda celulei este îngreunată de foarte multe ori și de faptul că fiecare tip de echipament (NC, CNC, ROC, PLC) utilizează în general limbaje de comandă diferite. Mai mult, poate să existe diferențe de limbaj chiar și între două echipamente de același tip (de ex. CNC) dar de proveniență diferită.

2.3.2. Structură de comandă cu interconectare printr-o rețea locală LAN

În această variantă atât controlerul de celulă cât și fiecare echipament de comandă din cadrul celulei este conectat la o rețea locală LAN (Local Area Network). Având în vedere că aceasta din urmă se află în cadrul unei celule de fabricație, este numită **Rețea Locală de Celulă** (Cell Local Area Network) sau simplu **Rețea de Celulă** (Cell Network).

Pentru conectarea controlerului de celulă respectiv a echipamentelor de comandă la Rețeaua de Celulă, ele trebuie să fie dotate cu așa numite **interfețe LAN**.

Observație:

- O interfață LAN are ca funcție principală, codificarea informațiilor transmise prin mediul fizic al rețelei LAN, respectiv decodificarea informațiilor recepționate.

Dacă din motive constructive, echipamentele de comandă dintr-o celulă nu pot fi dotate cu interfețe LAN, atunci ele trebuie să fie conectate (serial) la așa numite **“calculatoare de interfață cu rețeaua”** care la rândul lor sunt dotate cu aceste interfețe, fig.2.5., [97].

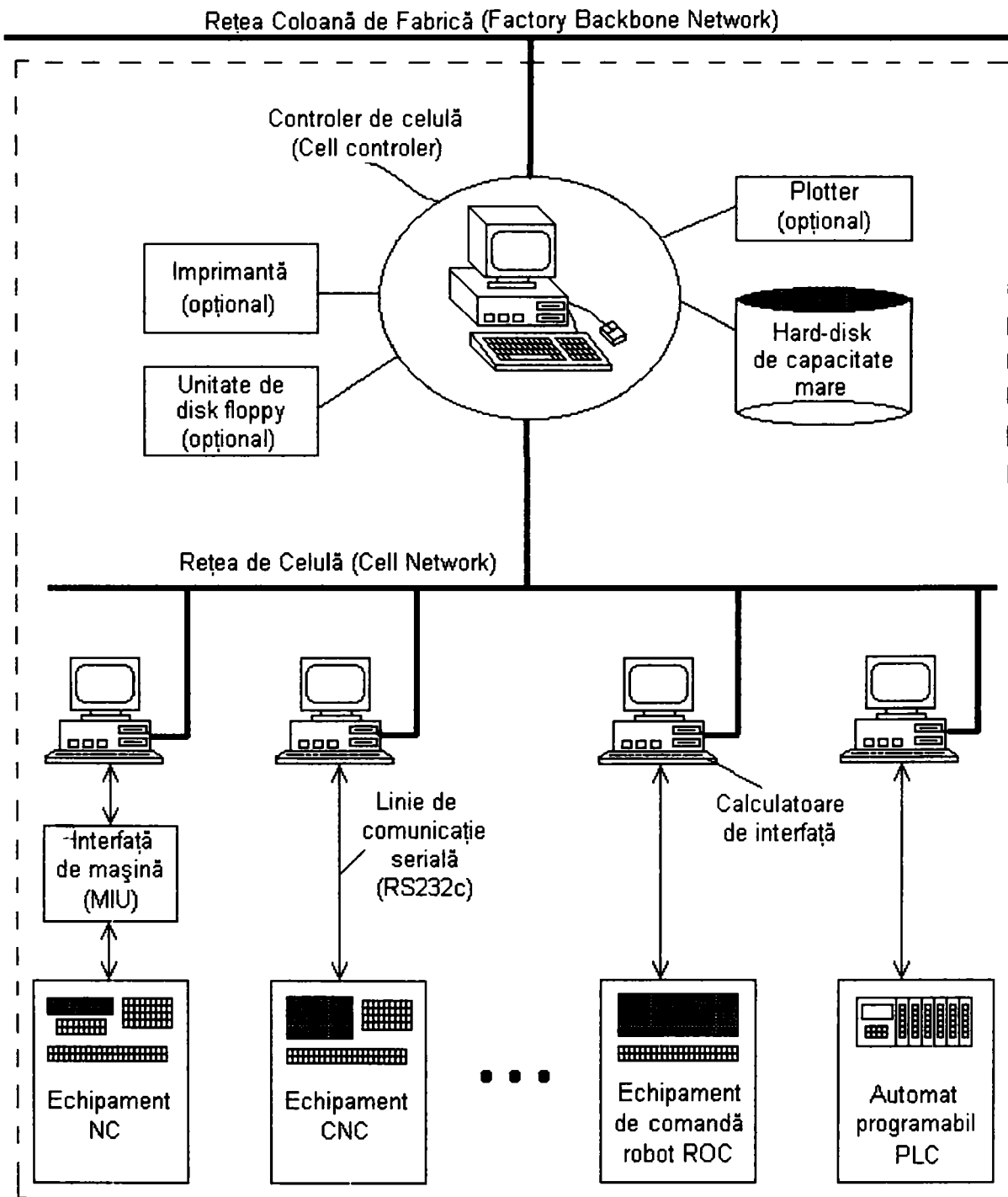


Fig.2.5. Modelul conectării la Rețeaua de Celulă prin intermediul unor “calculatoare de interfață”

208.057
268

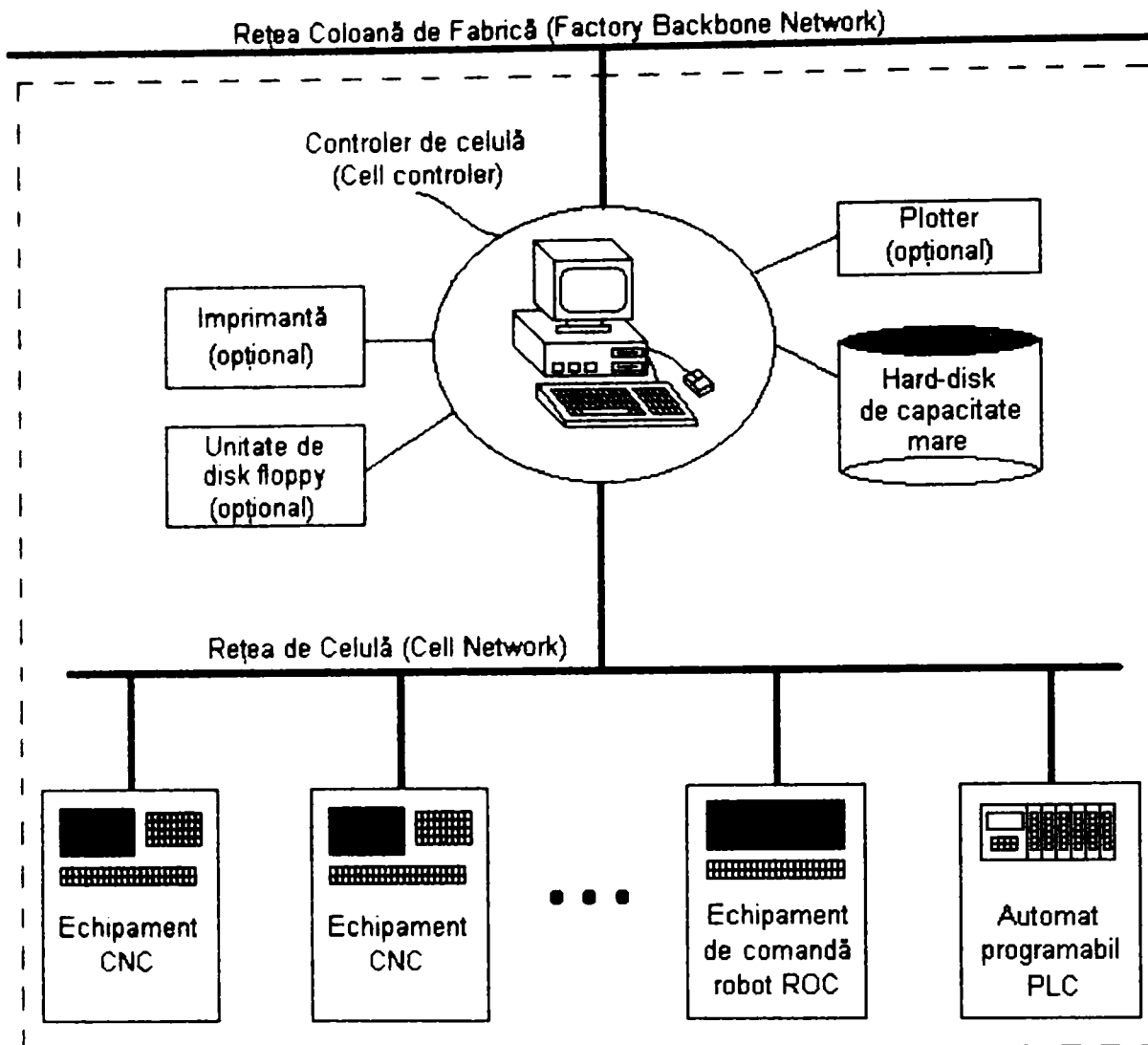


Fig.2.6. Modelul conectării "directe", la Rețeaua de Celulă (fără calculatoare de interfață)

Noile echipamente de comandă (CNC, ROC și PLC) sunt dotate, în general încă din fabricație, cu interfețe LAN. Ca urmare, aceste echipamente pot fi conectate la Rețeaua de Celulă direct, adică fără acele calculatoare de interfață, fig.2.6. Singura condiție este ca interfețele LAN să fie identice sau compatibile d.p.d.v. hardware și software.

În cazul în care Controlerul de Celulă este conectat și la așa numită **Rețea Coloană de Fabrică** (Factory Backbone Network), atunci există posibilitatea integrării a activităților de fabricație, nu numai la nivelul unei celule de fabricație, ci și la nivelul halei industriale.

Observație:

- Rețeaua Coloană de Fabrică străbate, sub forma unei "**coloane vertebrale**", întreaga hală industrială a fabricii, realizând astfel interconectarea informațională a rețelelor de celulă, fig.2.7., respectiv conectarea lor la **Controlerul Principal de Uzină** (Host Plant Controller).
- Conectarea rețelelor de celulă la rețeaua coloană de fabrică se poate realiza fie direct prin controlerul de celulă, fie prin utilizarea unor echipamente speciale de interconectare (Bridge, Router, Gateway etc.).

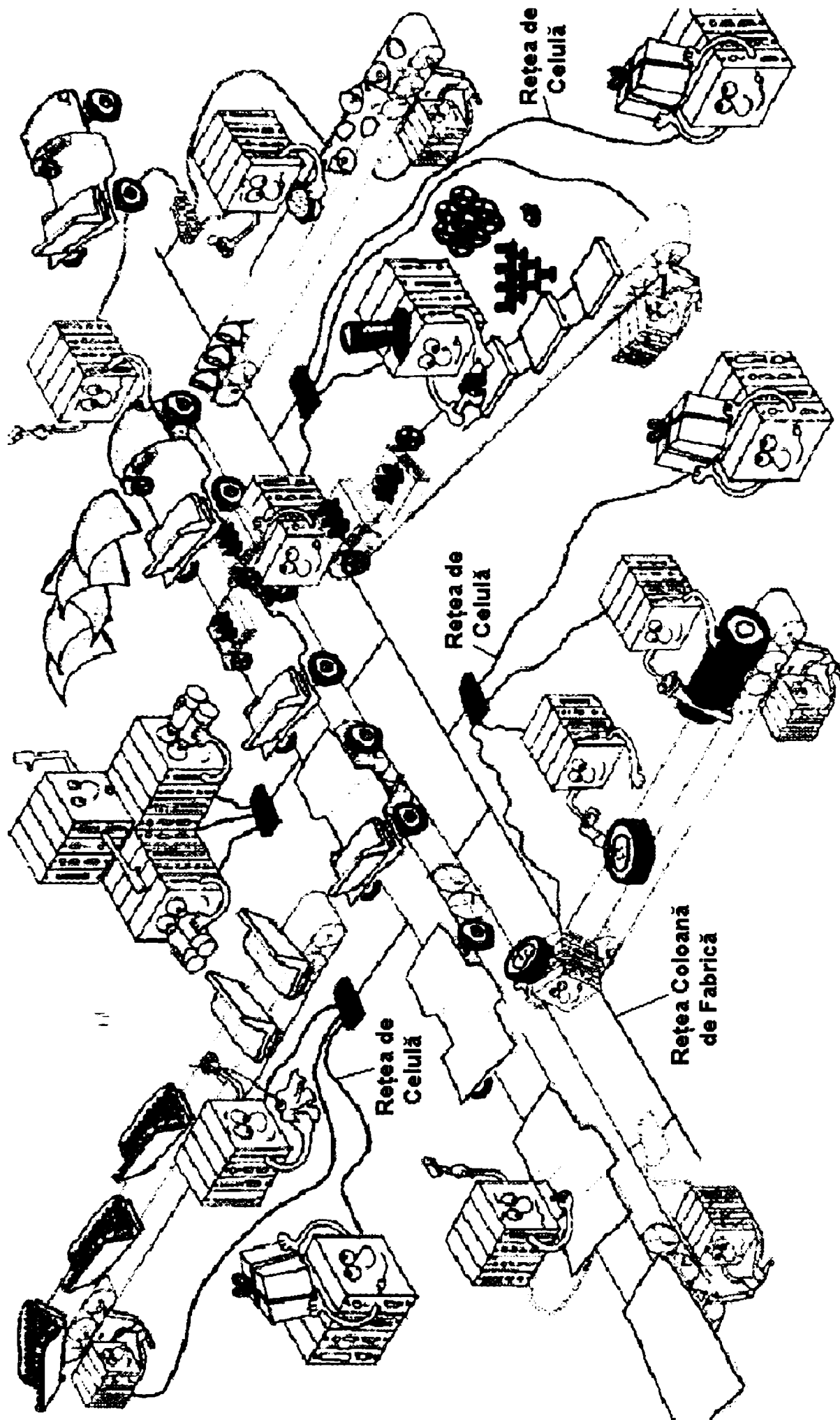


Fig.2.7. Exemplu de interconectare la nivelul halei industriale a Rețelelor de Celulă prin Rețeaua Coloană de Fabrică, [48]

Avantajul utilizării rețelelor locale LAN pentru efectuarea schimburilor informaționale derivă din faptul că aceasta oferă viteze de comunicație mult mai mari decât interconectarea serială RS232c, respectiv oferă posibilitatea efectuării unor schimburi directe de informații (nu neaparat prin intermediul controlerului de celulă).

Utilizarea unei rețele de celulă, proiectată corespunzător, este condiția esențială pentru realizarea **coordonării în timp real a activităților de fabricație** din cadrul celulei de fabricație.

În fig.2.8. s-a încercat ilustrarea conectării directe a echipamentelor de comandă la Rețeaua de Celulă de tip LAN.

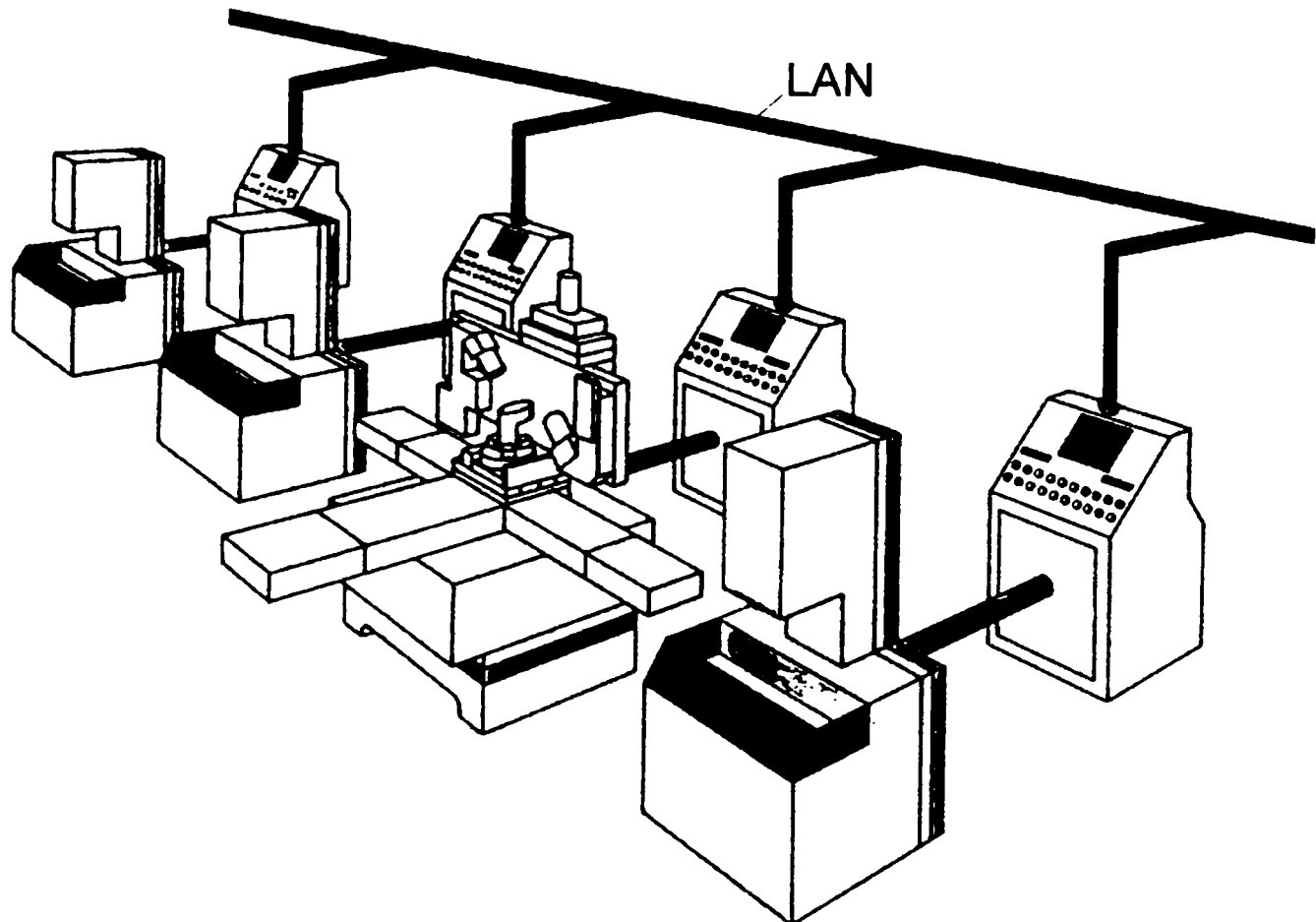


Fig.2.8. Ilustrarea conectării directe a echipamentelor la rețeaua de celulă

În fig.2.9. s-a reprezentat un exemplu de interconectare a echipamentelor de comandă din cadrul unei celule de fabricație. Acesta conține în plus, pe lângă echipamente de comandă numerică (CNC), echipamente de comandă robot (ROC), automate programabile (PLC) și respectiv controlerul de celulă și o **stație de lucru** (workstation) formată dintr-un calculator PC. (În fig.2.8. nu s-a reprezentat legătura cu rețeaua coloană de fabrică.)

Stația de lucru este utilizată uneori pentru monitorizarea și respectiv urmărirea activităților de fabricație din cadrul celulei, de către un operator uman. Eventualele intervenții sunt efectuate în scopul optimizării programului de fabricație. De asemenea, operatorul uman obține rapid informații referitoare la situația producției neterminate.

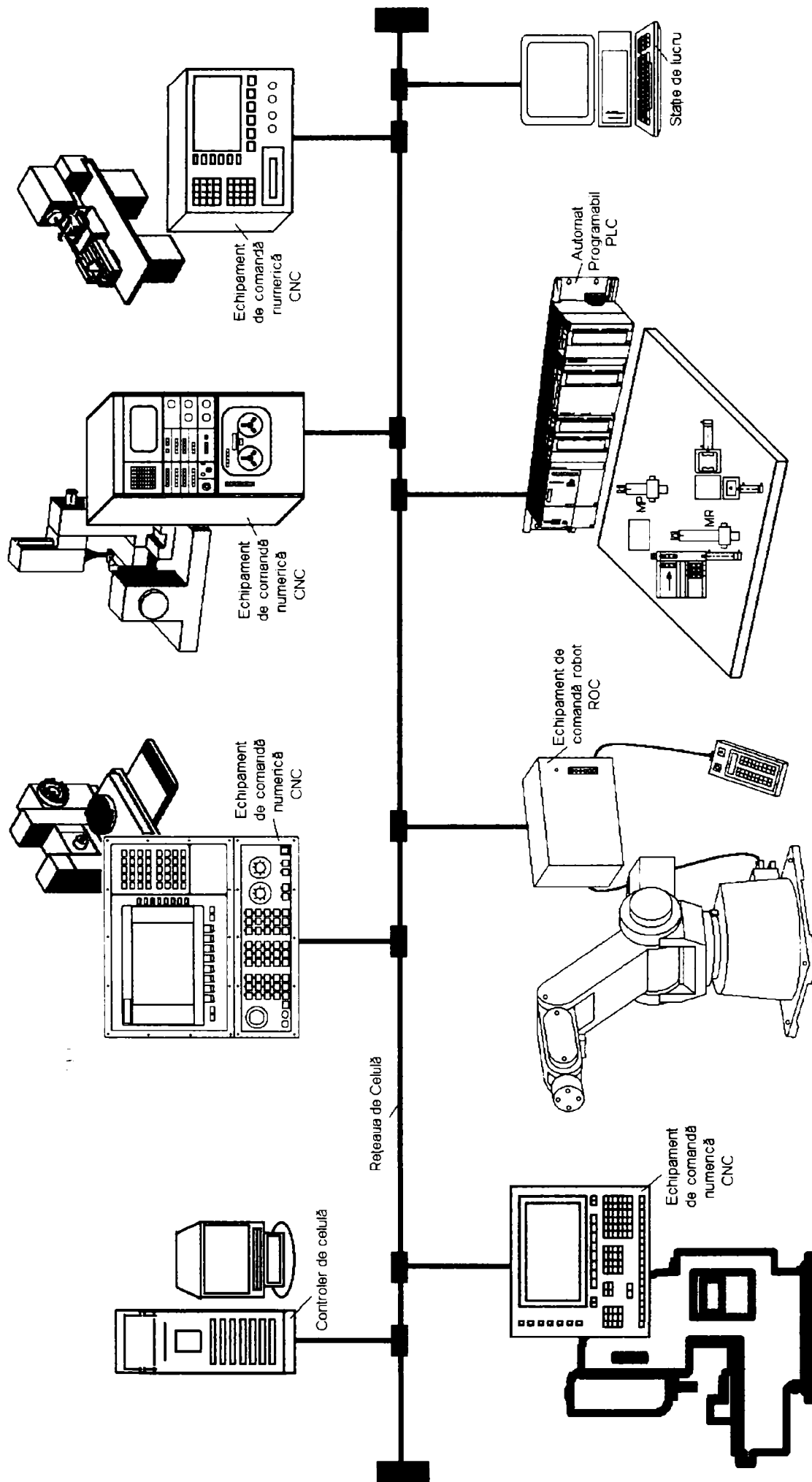


Fig.2.9. Exemplu de interconectare a echipamentelor de comandă din cadrul unei celule de fabricație, [6]

2.4. Structuri tipice de automatizare a fabricației la nivelul compartimentelor de execuție al unui Hipersistem CIM

În fig.2.10. s-a încercat reprezentarea modelului unei întreprinderi, din domeniul construcțiilor de mașini, care este organizată conform principiilor Hipersistemului CIM, [6]. Pe baza figurii pot fi distinse ușor patru nivele de activitate, [69, 71]:

- Nivelul 1. Conducere strategică.
- Nivelul 2. Compartimente funcționale (servicii).
- Nivelul 3. Compartimente de concepție.
- Nivelul 4. Compartimente de execuție.

Activitățile de fabricație din cadrul întreprinderii, respectiv mijloacele de automatizare ale acestora fac parte din nivelul 4., adică din **nivelul compartimentelor de execuție**. Compartimentele de execuție care sunt strâns legate de automatizarea fabricației sunt următoarele:

- compartimentele de debitare, de fabricație respectiv de finisare, care au în componența lor una sau mai multe **Celule de Fabricație Flexibilă** (Flexible Manufacturing Cells - FMC);
- compartimentul de montaj, cu una sau mai multe **Celule de Asamblare Flexibilă** (Flexible Assembling Cell – FAC);
- compartimentul de control și testare, respectiv **Celula de Măsurare și Control de Calitate** (Measuring and Quality control Cell);
- compartimentul de depozitare cu un **Sistem Automat de Depozitare și Regăsire** (Automated Storage & Retrieval System – AS&RS);
- transportul uzinal, format dintr-un **Sistem de Vehicule Ghidate Automat** (Automated Guided Vehicle System - AGVS).

Celulele de Fabricație Flexibilă (Flexible Manufacturing Cells – FMC), sunt destinate operațiilor tehnologice de debitare, fabricație și respectiv finisare a semifabricatelor. Ele fac parte din compartimentele de debitare, fabricație respectiv de finisare. În cadrul acestor celule, modulele componente sunt organizate în jurul unor mașini de lucru, care sunt prevăzute cu sisteme de transfer automat a pieselor sau sunt servite de roboți industriali, [71]. **Structurile tipice de automatizare** legate de comanda celulei au fost prezentate în paragraful 2.3. Astfel comanda celulei (adică coordonarea operațiilor de fabricație din cadrul celulei), se realizează prin intermediul unui **controler de celulă**, la care se conectează direct sau prin intermediul unei rețele locale LAN celelalte echipamente de comandă din cadrul celulei.

Celulele de Asamblare Flexibilă sunt destinate operațiilor de montaj respectiv de asamblare automată a componentelor executate. Organizarea și respectiv **comanda celulei** de asamblare flexibilă este similară cu cea a celulei de fabricație flexibilă, adică se realizează tot prin intermediul unui controler de celulă, însă modulele componente sunt organizate în jurul unor posturi de asamblare (în locul mașinilor de lucru). Mijlocul de producție principal în aceste celule este **robotul industrial**. Acesta, sau un grup de roboți sunt utilizați pentru efectuarea unor mișcări de manipulare complexe care au ca rezultat final montajul componentelor produsului finit.

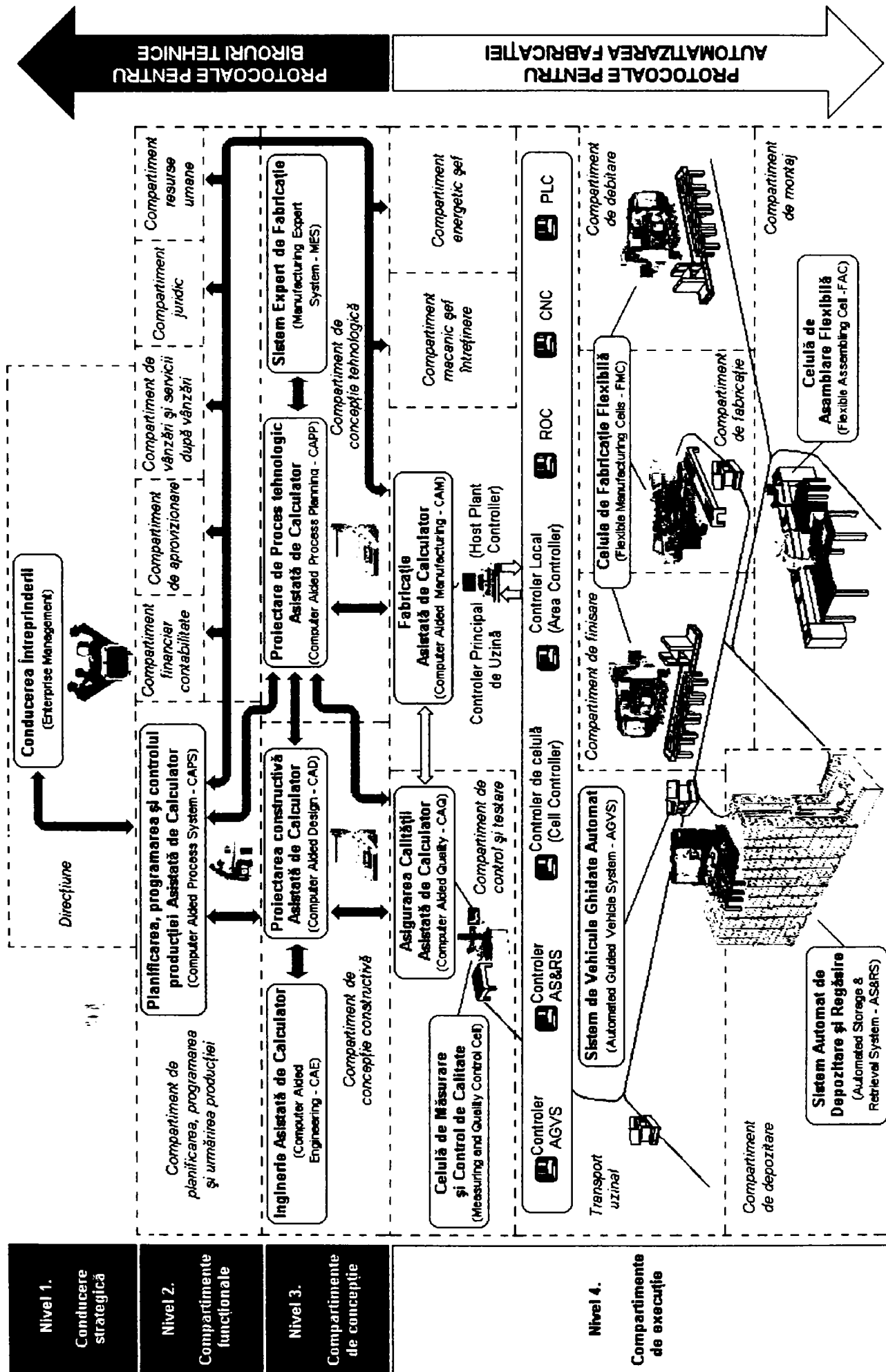


Fig.2.10. Modelul unei întreprinderi organizat pe baza principiilor Hipersistemului CIM

Celulele de Măsurare și Control de Calitate sunt destinate operațiilor de control și/sau de testare a componentelor/ansamblurilor executate. În cadrul acestor celule, controlul automat este de tip postoperație (postcelulă), el se realizează după efectuarea unei operații din cadrul procesului tehnologic. Pentru controlul automat se utilizează în general **roboți de măsurare** sau **mașini de măsurare cu comandă numerică**. Aceștia determină în mod automat de regulă dimensiuni dar pot fi utilizate și pentru controlul calității suprafeței obiectului de lucru, incertitudinea de execuție a unor operații de asamblare (spre exemplu suduri) sau pentru verificarea existenței/inexistenței unor defecte de material (spre exemplu fisuri), [71].

Operațiile de control automat pot fi realizate și prin intermediul unor **sisteme de vedere artificială**, utilizând în acest scop camere video (CCD). Prin intermediul acestora este preluată imaginea obiectului controlat și comparată cu imaginea obiectului etalon. Așezarea obiectelor sub camera video se realizează tot automat, fie prin intermediul unui sistem de transfer automat al obiectelor fie prin intermediul unui robot industrial.

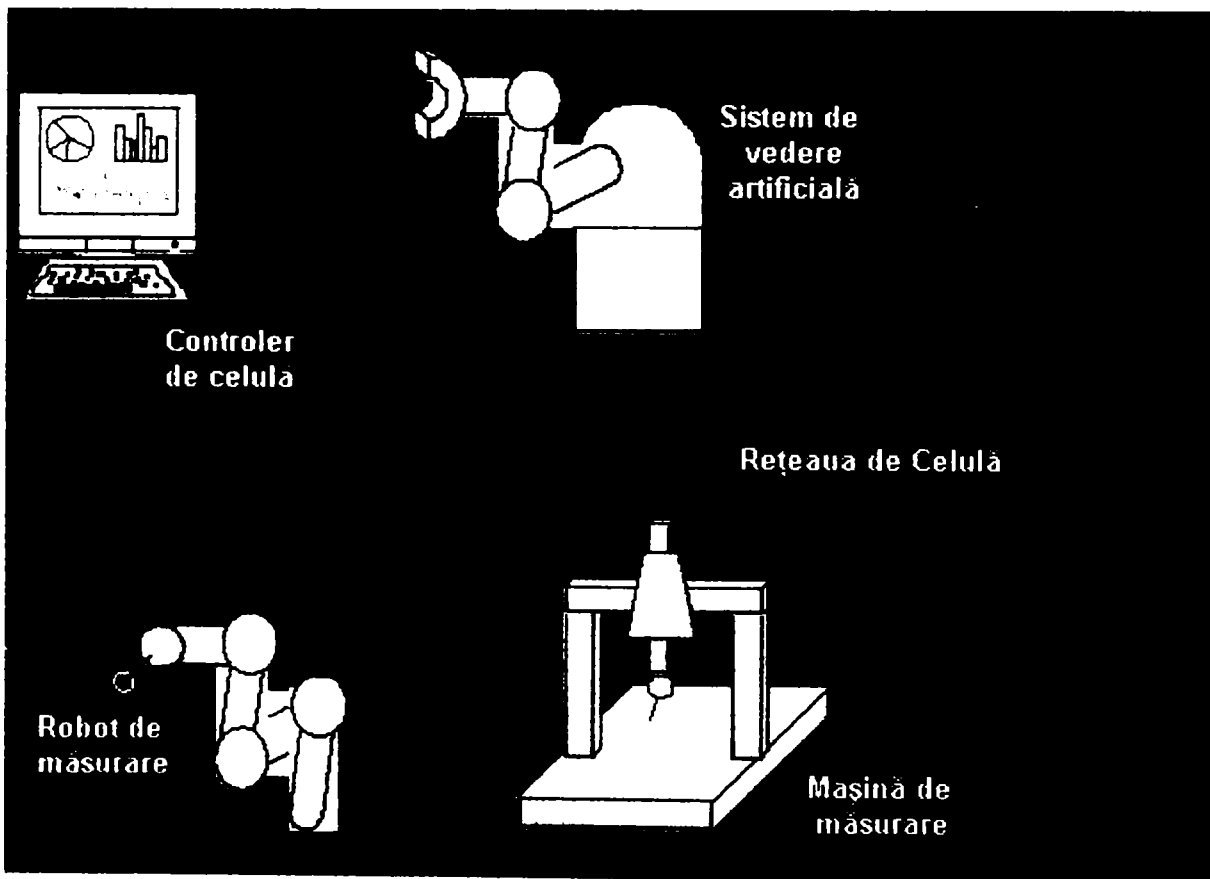


Fig.2.11. Componentele principale ale celulelor de măsurare și de control de calitate

În cadrul acestor celule pot fi realizate și operații de **testare automată**, [71]. Aceste operații presupun efectuarea unor funcții specifice de către obiectele de lucru, prelevarea unor date care descriu modul cum se realizează funcțiile în cauză, compararea acestor date cu cele memorate, corespunzătoare unei funcționări corecte și tragerea de concluzii favorabile/defavorabile cu privire la calitatea obiectului testat.

Structura sistemului de comandă este similară cu cea a celulei de fabricație flexibilă, fig.2.11. Însă controlerul de celulă, prin coordonarea operațiilor de control respectiv de testare automată realizează și funcția de **Asigurarea Calității Asistată de Calculator**.

Sistemul Automat de Depozitare și Regăsire (Automated Storage and Retrieval System – AS&RS) este destinat operațiilor de:

- *depozitare a materiei prime*, materialelor, pieselor, subansamblurilor, componentelor necesare pentru începerea procesului de fabricație,
- *depozitare intermediară* între fazele de intrare/operații ale semifabricatelor,
- *depozitare finală*.

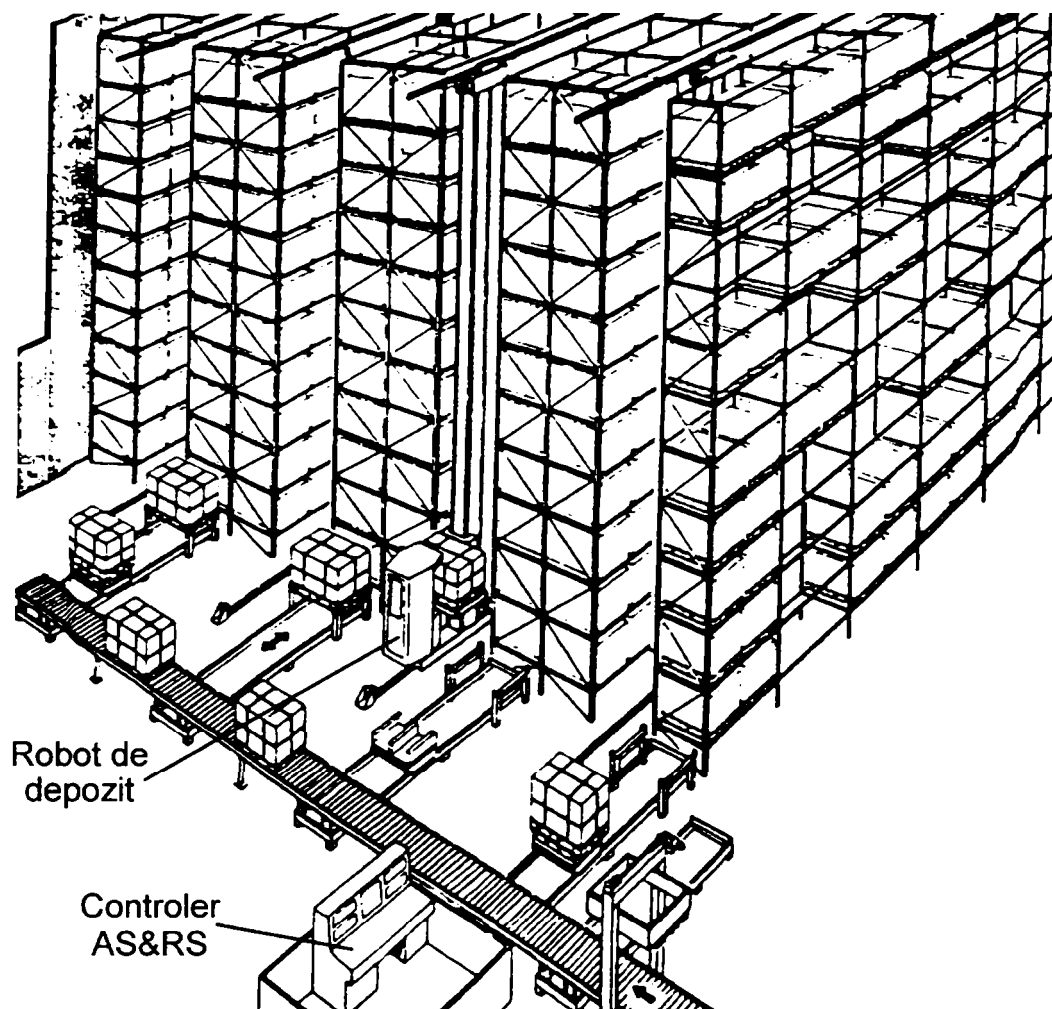


Fig.2.12. Exemplu de sistem automat de depozitare și regăsire

Depozitul automat (care poate fi organizat ca depozit cu rafturi înalte, cu compartimente denumite “cușete” sau ca depozit cu “casete autoportante”), este servit de **roboți de depozit**. În “cușete”, respectiv “casete” sunt depozitate materii prime, materiale, semifabricate, componente, subansambluri, depozitate individual sau așezate în palete și containere. Evidența acestora (natura, cantitatea/“cușetă” sau “casetă”) este memorată de “calculator” denumit **controler AS&RS**. Acesta, pe lângă gestiunea depozitului, mai are și funcția de coordonare a roboților de depozit prin legături informaționale cu echipamentele de comandă ale acestora. **Structura sistemului de comandă** poate fi asemănătoare celor prezentate la celule de fabricație, însă lipsesc echipamentele NC/CNC, iar locul controlerului de celulă este preluat de controlerul AS&RS.

Transportul uzinal al obiectelor de lucru între depozit și celule de fabricație, de asamblare respectiv de măsurare se realizează prin sistemul de transfer (transport) automat, materializat în majoritatea cazurilor prin **Sistemul de Vehicule Ghidate Automat** (Automated Guided Vehicle System – AGVS). Acesta este format

din vehiculele ghidate automat (AGV), calea de ghidare, stații de încărcare/descărcare, stații de încărcare acumulatori, sistem de comandă, [98].

Vehiculele ghidate automat (robocare) sunt vehicule acționate electric, conduse de un calculator imbarcat, care este în legătură informațională (în anumite puncte de comunicație de pe traseu) cu un calculator central de sistem numit **Controler AGVS**, fig.2.13., [97]. Vehiculele sunt prevăzute cu acumulator de energie și cu dispozitive de încărcare/descărcare automată a pieselor/paletelor.

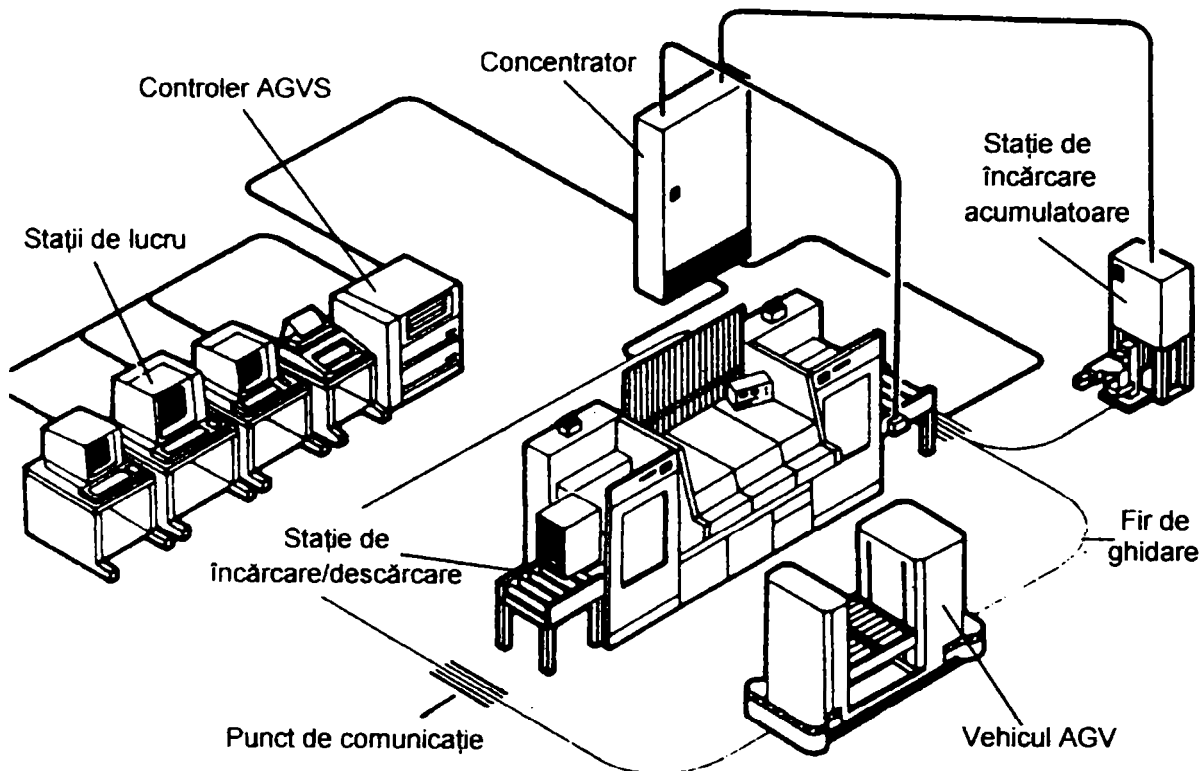


Fig.2.13. Elementele componente ale sistemului de vehicule ghidate automat AGVS

Ghidarea automată a vehiculelor se realizează de regulă prin intermediul unui fir de ghidare îngropat, prin care este vehiculat un curent electric de înaltă frecvență. Câmpul magnetic generat de fir este detectat de două bobine de dirijare, situate în fața vehiculului. Semnalele induse în bobine sunt amplificate, filtrate și comparate între ele, obținându-se un semnal de eroare proporțional cu deviația vehiculului în raport cu traseul. Acest semnal este folosit pentru comanda sistemului de direcție al vehiculului în vederea revenirii automate pe traseu. Ciocnirea vehiculelor cu obstacole de pe traseu este prevenită prin intermediul unor senzori ultrasonici. Acești senzori realizează un ecran de protecție la circa 1,5m distanță de vehicul, determinând reducerea vitezei, urmată de oprire dacă obstacolul nu dispare. Dacă acesta este înlăturat, pornirea vehiculului se face automat.

Vehiculele ghidate automat realizează încărcarea/descărcarea obiectelor/paletelor la posturile de încărcare/descărcare fixe ale depozitului AS&RS, ale celulelor de fabricație, de asamblare respectiv de măsurare.

Comanda sistemului de vehicule ghidate automat se realizează de către **Controlerul AGVS**. Acesta este în legătură informațională cu vehiculele, stațiile de încărcare/descărcare respectiv cu stația de încărcare acumulatori prin intermediul unui echipament numit **Concentrator**.

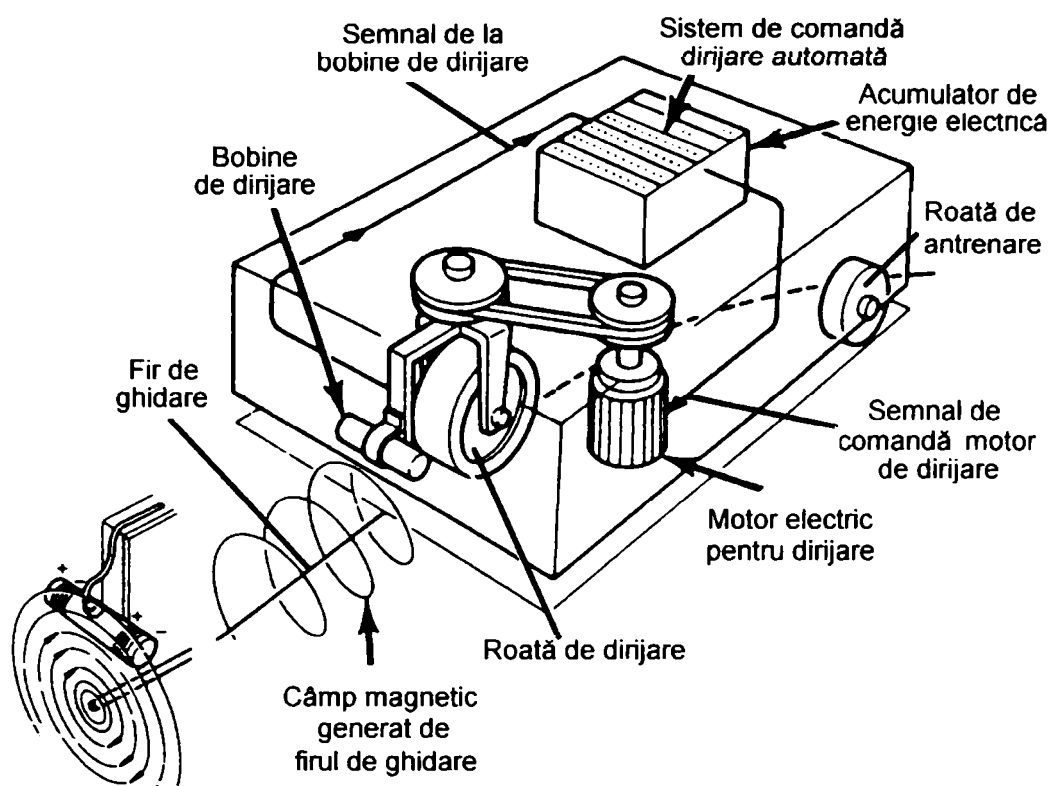


Fig.2.14. Principiul fundamental al dirijării automate a vehiculelor AGV, [97]

Controlerul AGVS comandă încărcarea/descărcarea vehiculelor ghidate automat în anumite posturi cu semifabricate, piese, subansambluri, ansambluri, separate sau aflate pe palete, traseele pe care circulă, staționarea lor, încărcarea acumulatorilor la posturi de alimentare cu energie electrică. Schimbul de informații între calculatorul îmbarcat pe vehiculul ghidat și Controlerul AGVS se realizează în posturi speciale de schimb de informații (puncte de comunicație) sau prin legături radio.

Coordonarea activităților de fabricație din cadrul aceleiași atelier sau secții se realizează în general prin intermediul unor "calculatoare" numite **Controlere Locale** (Area Controller). La acesta sunt conectate informațional, toate controlerile de celulă din cadrul atelierului sau secției.

Coordonarea tuturor activităților de fabricație de la nivelul 4. al întreprinderii (nivelul compartimentelor de execuție), este realizată prin intermediul **Controlerului Principal de Uzină** (Host Plant Controller). La acesta sunt conectate informațional, controlerile locale, controlerul AS&RS respectiv controlerul AGVS. Interconectarea controlerilor de celulă, locale, AS&RS, AGVS respectiv a controlerului principal de uzină se realizează în general prin intermediul Rețelei Coloană de Fabrică. Tot la această rețea sunt conectate și *stațiile de lucru* de la nivelul halei industriale a fabricii (Factory Workstations), care pot fi formate din mini-calculatoare (Mini-Computers) cu mai multe terminale și/sau din Calculatoare Personale (PC). O parte din aceste stații pot fi situate chiar și în interiorul unor celule de fabricație (Cell Workstations) caz în care ele sunt conectate direct la rețeaua de celulă (Cell Network).

Pe baza celor de mai sus s-a realizat **modelul general al structurii de comandă aferent nivelului 4 al Hipersistemului CIM**, reprezentat în fig.2.15.

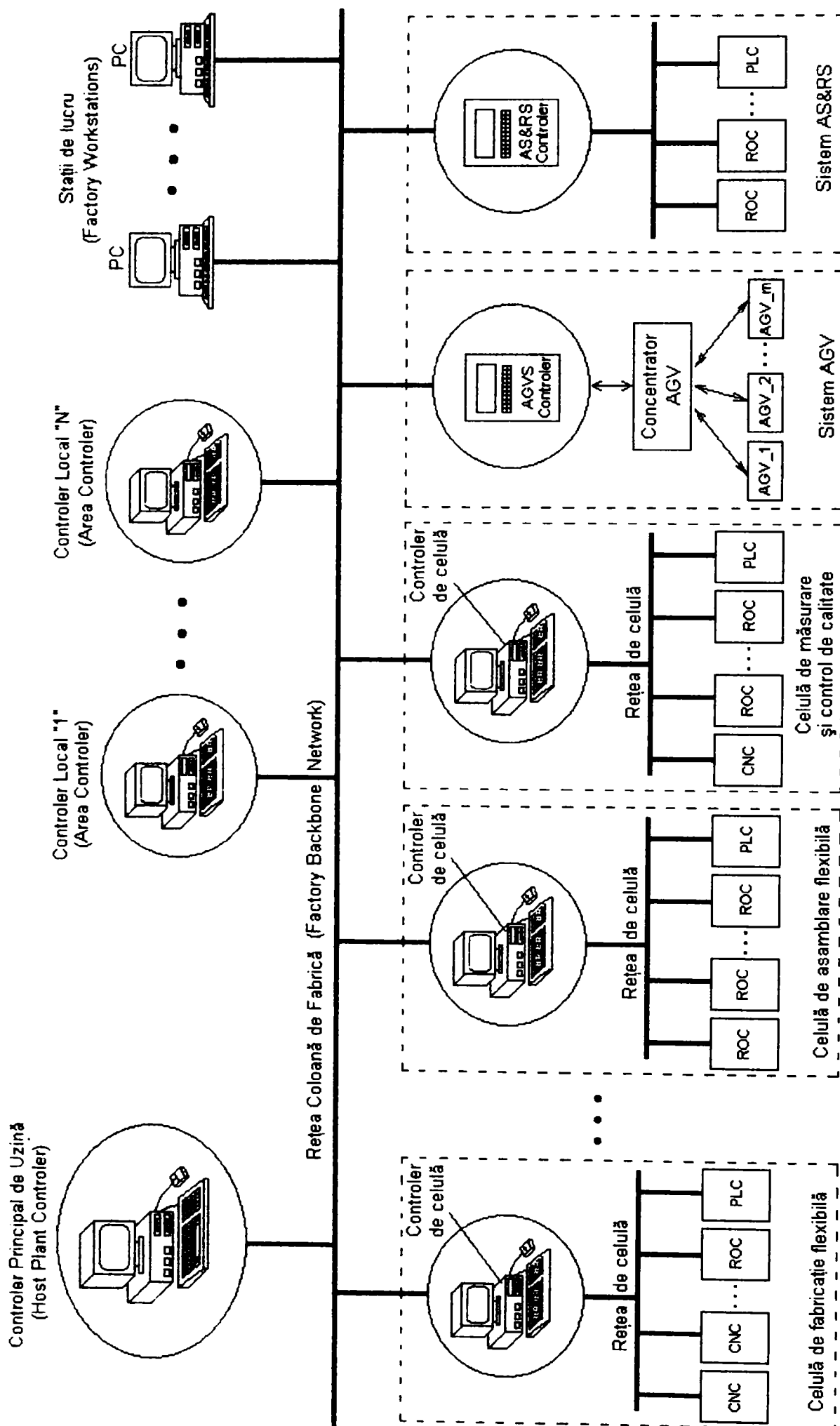


Fig2.15. Modelul general al structurii de comandă aferent nivelului 4 al Hipersistemului CIM, [6]

Observații:

- Totalitatea activităților de comandă/coordonare, realizate prin intermediul echipamentelor de comandă (CNC, ROC, PLC), controlerele de celulă, de AS&RS, de AGVS, de controlerele locale respectiv de controlerul principal de uzină este realizată prin participarea activă a “calculatorului”. Astfel totalitatea acestor activități de comandă/coordonare formează modulul funcțional de **Fabricație Asistată de Calculator** (Computer Aided Manufacturing - CAM) al Hipersistemului CIM.
- Cu cât este mai mare nivelul tehnic de implementare a funcțiilor de comandă/coordonare din cadrul acestui modul CAM, cu atât este mai mare **gradul de automatizare a fabricației** din cadrul Hipersistemului CIM.

CAPITOLUL 3.

ANALIZA EVOLUȚIEI STRUCTURILOR DESGHISE DE COMANDĂ DIN CADRUL HIPERSISTEMELOR CIM (“OPEN CONTROL”)

3.1. Introducere în comanda de la distanță a echipamentelor de fabricație

Înainte de prezentarea problemelor care au condus la introducerea în Hiper-sisteme CIM, a structurilor deschise de comandă, trebuie să definim și să analizăm modul de realizare a comenzii de la distanță a echipamentelor de fabricație.

3.1.1. Ce este comanda de la distanță a unui echipament de fa- bricație integrat într-un Hipersistem CIM?

Prin comanda unui echipament de fabricație (mașină de lucru, robot industrial etc.) se înțelege acele acțiuni umane și/sau automate prin care se **realizează accesul la controlul resurselor fizice și logice** ale acestuia. Aceste acțiuni au ca scop punerea în funcțiune, respectiv modificarea și/sau monitorizarea stării de funcționare a echipamentului respectiv.

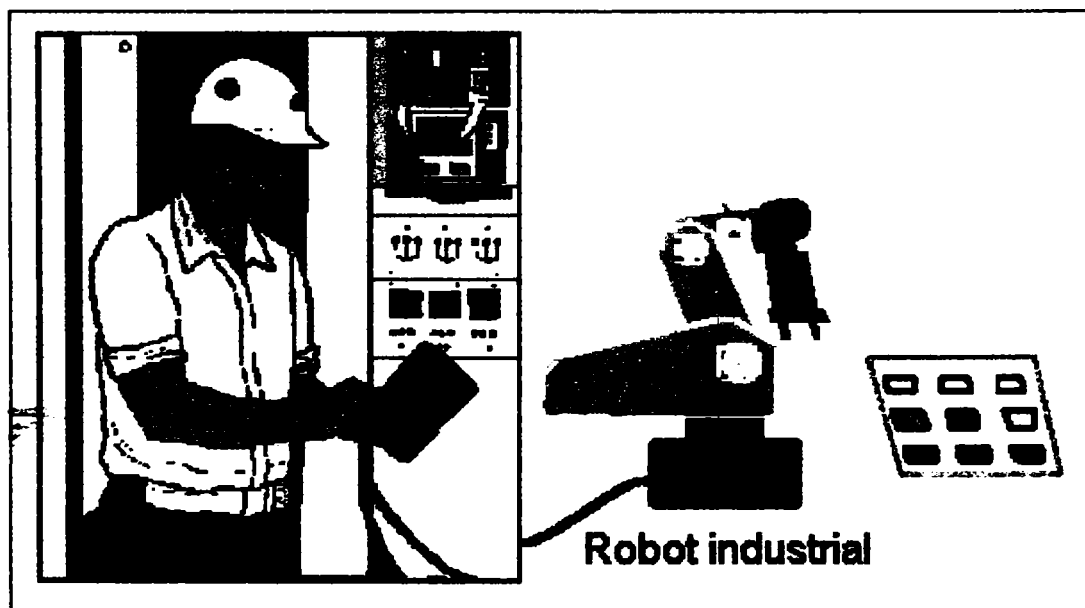


Fig.3.1. Comanda “locală” a unui robot industrial

Dacă aceste acțiuni nu sunt rezultatul unor legături cu rețeaua informațională a Hipersistemului CIM, atunci ele realizează un control local al funcționării echipamentului de fabricație. Acțiunile prin care se realizează acest control local se numesc **comenzi locale** (local control).

Un exemplu pentru comandă locală poate fi acțiunea unui operator uman care prin intermediul unei console de programare, panou de instruire sau echivalent, reali-

zează punerea în mișcare a unui robot industrial, fig.3.1. La fel, o comandă locală poate fi și rezultatul unor acțiuni automate (care au loc prin intermediul unui echipament local de comandă respectiv fără intervenția operatorului uman) și concură de asemenea la punerea în mișcare a robotului.

Prin **comanda de la distanță** (remote control) a unui echipament de fabricație, integrat într-un Hipersistem CIM, vom înțelege acele acțiuni umane și/sau automate prin care se realizează un control “de la distanță” al funcționării acestuia, adică prin rețeaua informațională a Hipersistemului. Prin aceste acțiuni (în continuare comenzi) se realizează de fapt **accesul “de la distanță” la controlul resurselor fizice și logice** ale echipamentului respectiv.

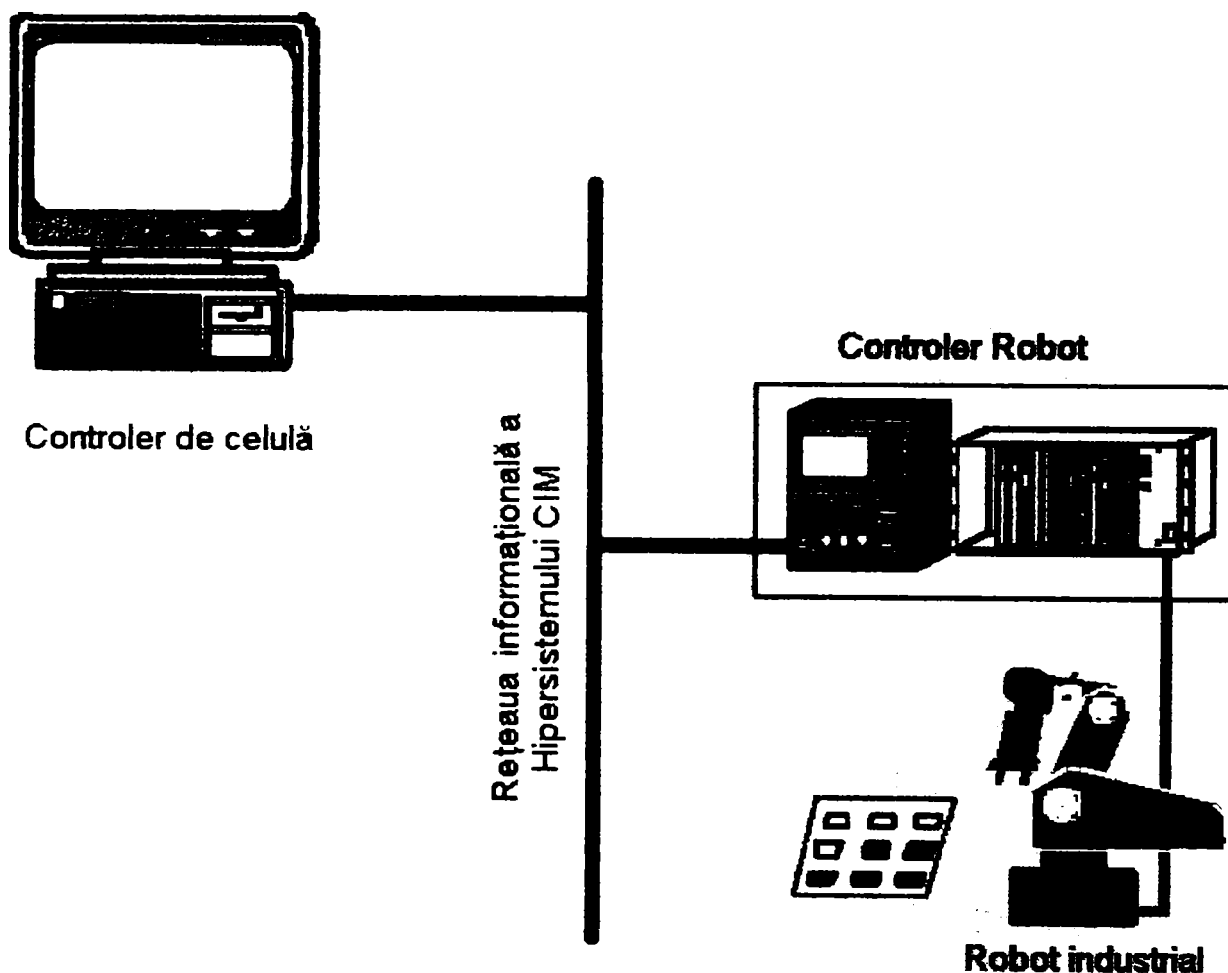


Fig.3.2. Comanda “de la distanță” a unui robot industrial (prin rețeaua informațională a Hipersistemului CIM)

Un exemplu pentru comanda de la distanță poate fi controlul funcționării unui robot industrial cu ajutorul unor “comenzi” emise, prin rețeaua informațională a Hipersistemului CIM (de exemplu prin Rețeaua de Celulă), de către un calculator aflat pe rol de controler de celulă, fig.3.2.

Observații:

- Este de menționat faptul că realizarea comenzii de la distanță, **nu are ca scop preluarea** de către un calculator “îndepărat”, a **controlului nemijlocit** al echipamentului de fabricație (de exemplu a controlul nemijlocit al mișcărilor la un robot industrial). Acesta se rezolvă totdeauna prin mijloace “locale” specializate.

- Însă se urmărește efectuarea unui control de la distanță asupra stărilor de funcționare (pornire, oprire, încărcarea și lansarea unui program de aplicație etc.) în vederea realizării **sincronizării cu funcționarea altor echipamente de fabricație** (alți roboți, mașini-unelte etc.), integrate de asemenea în structurile Hipersistemului CIM. Acesta este de fapt scopul realizării comenzii de la distanță.

3.1.2. Modul de realizare a comenzii de la distanță

Comanda de la distanță a unui echipament de fabricație, integrat într-un Hipersistem CIM, se realizează prin **emiterea către acesta a unor mesaje informaționale**, care sunt recunoscute de către echipamentul respectiv, respectiv **sunt interpretate drept comenzi** și ca urmare sunt executate.

Observație:

- În cadrul comenzii de la distanță se realizează și o **monitorizare a stărilor de funcționare** ale echipamentului respectiv. Aceasta este posibilă, tot prin intermediul unor mesaje informaționale, care însă sunt emise de această dată de echipamentul de fabricație.

Întregul schimb de mesaje informaționale se realizează prin rețeaua informațională a Hipersistemului CIM, respectând o **arhitectură client-server**. Pentru înțelegerea realizării modului de schimb de mesaje informaționale, între două echipamente de comandă, de exemplu între un controler de celulă și un echipament de comandă de tip CNC, ROC sau PLC, să considerăm modelul schimbului de informații între două "stații" ale Hipersistemului CIM.

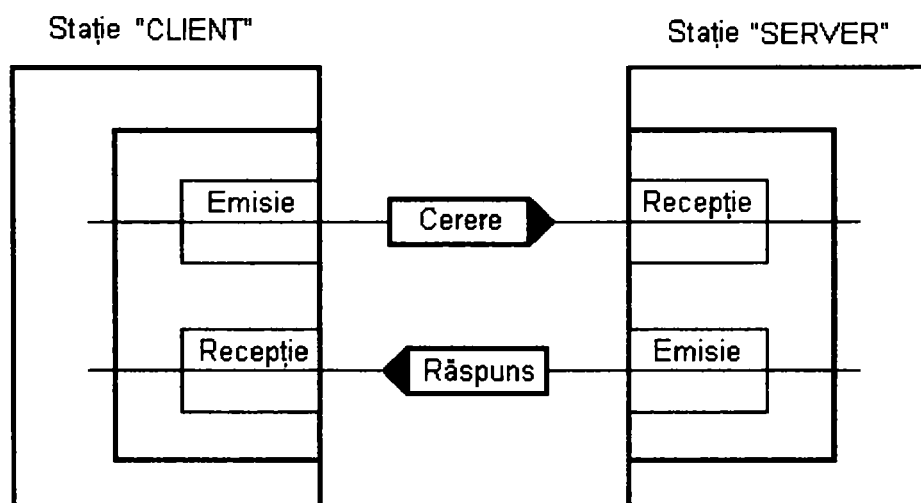


Fig.3.3. Modelul client-server, de schimb de informații între două stații, [23]

Pe baza modelului, (fig.3.3.), o stație pe rol de CLIENT emite (sending) către o altă stație, care este pe rol de SERVER, o cerere (request) privind efectuarea de către aceasta a unui serviciu. SERVER-ul recepționează (receiving) această "**cerere de serviciu**" (service request) înaintată de CLIENT și verifică dacă o poate efectua sau nu. În caz afirmativ efectuează serviciul cerut și emite înapoi către CLIENT un răspuns (response) pozitiv. În caz contrar emite un răspuns negativ, notificând și cauzele neefectuării serviciului cerut.

Pe rol de CLIENT (respectiv pe rol de SERVER) poate fi orice echipament de comandă din cadrul Hipersistemului CIM care este capabilă să emite (respectiv să răspunde la) cereri de servicii.

Observații:

1. În general, CLIENT-ul este cel care emite cererea iar SERVER-ul este cel care are sarcina de efectuare a serviciului solicitat în cadrul cererii.
2. Această arhitectură, utilizată de regulă în rețele de calculatoare, este cunoscută sub denumirea de "**arhitectură client-server**".

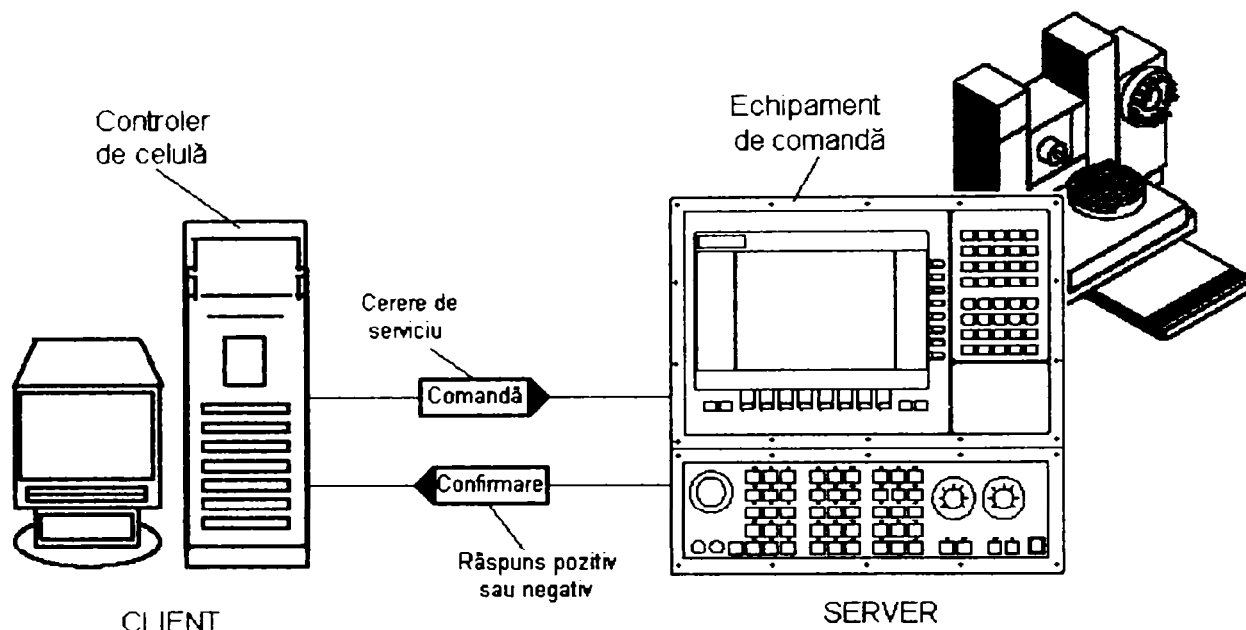


Fig.3.4. Relația client-server între controlerul de celulă și un echipament de comandă (de exemplu: CNC)

Într-o celulă de fabricație, pe rol de CLIENT se află controlerul de celulă. Acesta, pe baza programului de comandă a celulei de fabricație, la un anumit moment dat poate să emite o comandă, către un echipament de CNC, PLC sau ROC din cadrul celulei respective, (fig.3.4.). Comanda poate să fie de exemplu, "cererea" de lansare în execuție a unui program. În urma recepționării comenzii, mijlocul de producție (mașina de lucru, robotul industrial etc.), care se află sub comanda nemijlocită a echipamentului respectiv, va efectua o operație tehnologică (debitare, prelucrare, finisare, asamblare).

Echipamentul căruia îi este "adresată" comanda este pe rol de SERVER. Acesta va recepționa comanda și în prima fază va verifica dacă sunt îndeplinite sau nu toate condițiile (locale) efectuării comenzii respective. În caz afirmativ începe efectuarea ei, adică lansează în execuție programul specificat în cadrul comenzii. În paralel emite către controlerul de celulă (adică către CLIENT) o confirmare în care notifică faptul că comanda a fost efectuată, adică a fost începută execuția programului (operația tehnologică este în curs de derulare). Dacă comanda nu poate fi efectuată, de exemplu din cauza faptului că încă nu s-a terminat execuția programului curent (adică nu s-a terminat efectuarea comenzii anterioară), atunci emite către controlerul de celulă o confirmare în care notifică cauzele acesteia.

Comanda emisă de CLIENT (controlerul de celulă), către SERVER (un echipament CNC, ROC sau PLC din cadrul celulei), este de fapt o **cerere de serviciu**. Serviciul care trebuie să efectueze SERVER-ul este "lansarea în execuție a unui program" indicat în cadrul cererii (în cadrul comenzii). Confirmările: "**serviciul indicat poate fi efectuat**" (și în consecință este deja în curs de efectuare), respectiv "**serviciul nu poate fi momentan efectuat**", sunt de fapt răspunsurile (negative sau pozitive) ale SERVER-ului asupra cererii CLIENT-ului.

Cererea de serviciu emisă de controlerul de celulă respectiv răspunsurile (pozitive sau negative) emise de echipamentul de comandă (CNC, PLC sau ROC) din cadrul celulei se numesc în mod unitar **mesaje informaționale** sau simplu: **mesaje**.

În concluzie, comanda de la distanță a echipamentelor de fabricație, din cadrul Hipersistemului CIM, se realizează printr-un schimb de mesaje informaționale de tip client-server. Având în vedere că prin intermediul acestor mesaje se realizează în mod indirect și coordonarea activităților de fabricație de la nivelul celulelor (de fabricație), atelierelor (secțiilor) de producție și respectiv uzinei, ele mai sunt numite și "**mesaje de fabricație**" (manufacturing messages).

3.2. Cauzele inițiale care au condus la introducerea în Hipersisteme CIM a structurilor deschise de comandă

Cauzele inițiale care au condus la introducerea în Hipersisteme CIM a structurilor deschise de comandă derivă din modul de realizare a comenzii de la distanță. Practic fiecare echipament de comandă, care urmează să fie integrat într-un Hipersistem CIM, trebuie să îndeplinească simultan două criterii legate de comanda de la distanță:

1. **Să fie capabilă pentru emiterea și recepționarea mesajelor de fabricație** utilizate în cadrul Hipersistemului CIM. Aceasta este o cerință legată de utilizarea unei modalități comune de transmitere, prin rețeaua informațională a Hipersistemului CIM, a mesajelor de fabricație.
2. **Să fie capabilă să recunoască mesajele de fabricație** utilizate în cadrul Hipersistemului CIM **și să le interpreteze drept comenzi**. Aceasta este o cerință legată de utilizarea de către echipamentul respectiv a unui set comun de mesaje de fabricație cu cele utilizate în cadrul Hipersistemului CIM.

Dacă cel puțin una dintre cerințele enunțate mai sus nu este îndeplinită atunci integrarea informațională a echipamentului respectiv în acel Hipersistem CIM, este posibilă doar după o **prealabilă compatibilizare** a acestuia cu cele existente deja în cadrul sistemului. Compatibilizarea se realizează prin dotarea acestor echipamente cu interfețe hard și soft speciale, prin care sunt transmise și recepționate mesajele de fabricație, fig.3.5.

Aceste interfețe uneori pot fi echipamente de sine stătătoare, prin care echipamentul de comandă se conectează la rețeaua informațională a Hipersistemului CIM. În alte cazuri ele pot fi module speciale, care se montează în interiorul echipamentului de comandă. În ambele variante, ele sunt foarte scumpe. În consecință, integrarea într-un Hipersistem CIM a unui echipament de comandă care trebuie să fie

dotat cu astfel de "echipamente de compatibilizare", conduce inevitabil la costuri suplimentare de integrare informațională. Problema este că aceste costuri uneori sunt extrem de ridicate.

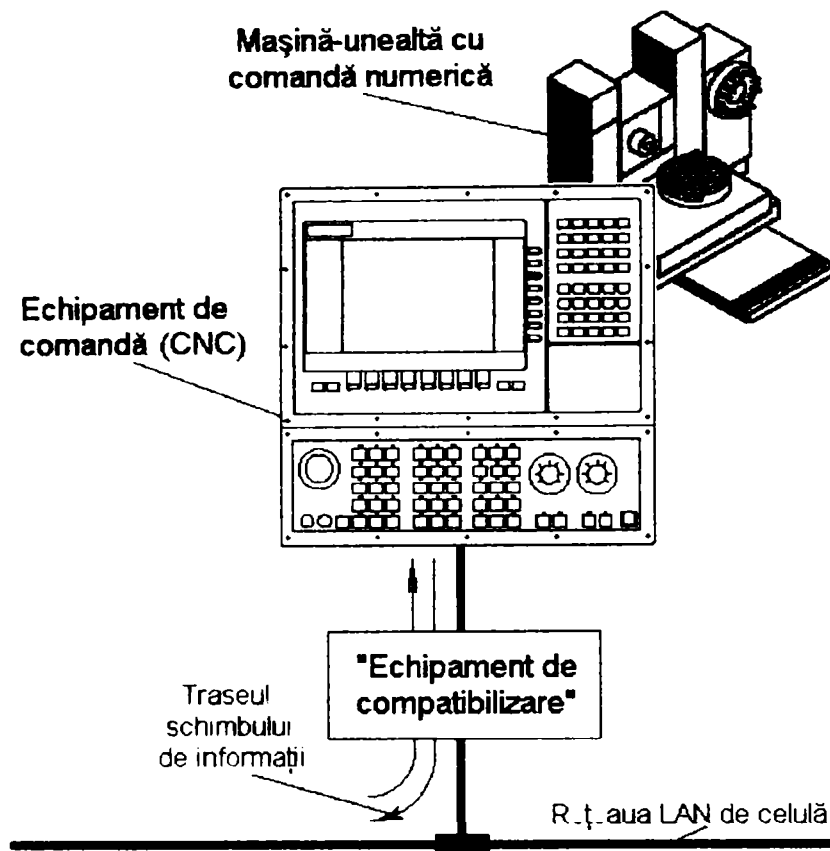


Fig.3.5. Conectarea unui echipament de comandă la rețeaua LAN de celulă prin intermediul unui echipament de compatibilizare

De exemplu, în anii '80 s-a ajuns într-o situație (este cazul firmei General Motors), în care s-a cheltuit aproape jumătate din prețul de achiziționare a echipamentelor de comandă pe compatibilizarea lor hard și soft, în vederea integrării lor în structuri de fabricație automatizate.

O modalitate pentru reducerea costurilor de integrare este **achiziționarea fiecărui echipament de comandă de la același producător**, care în general asigură compatibilitatea informațională a echipamentelor proprii. În acest caz nu mai este necesară utilizarea unor echipamente de compatibilizare. Pe de altă parte în cazul echipamentelor care provin de la același producător, este mai ușor de însușit procedeul de operare/programare (decât în cazul utilizării echipamentelor de proveniență diferită), [23].

Însă procesul de fabricație de regulă necesită achiziționarea mijloacelor de producție de diferite tipuri care în foarte multe cazuri sunt dotate cu echipamente de comandă provenite de la fabricanți diferiți.

Chiar dacă se încearcă totuși, pe cât posibil, achiziționarea unor mașini care să fie dotate cu echipamente identice cu cele existente deja în cadrul Hipersistemului CIM, totuși poate exista situația că acel model "vechi" de echipament nu mai este produs iar producătorul respectiv dorește mai mult vânzarea modelelor noi decât dezvoltarea unui model mai vechi.

În aceste condiții a devenit necesară elaborarea și introducerea în Hipersisteme CIM a unor **structuri deschise de comandă**. Aceste structuri sunt caracterizate prin faptul că, **specificația privind modul de realizare a comenzii de la distanță este făcută public**, adică orice utilizator sau fabricant are acces gratuit și nelimitat la ea. Scopul este ca majoritatea fabricanților de echipamente să implementeze în produsele lor proprii același mod comun de comandă de la distanță. Astfel structurile de comandă ale echipamentelor de fabricație, deși provin de la furnizori diferiți, vor fi compatibile d.p.d.v. informațional și în consecință vor genera costuri reduse la integrarea în Hipersisteme CIM.

3.3. Elaborarea și introducerea primelor structuri deschise de comandă în Hipersisteme CIM

Pentru elaborarea unor structuri deschise de comandă de la distanță este necesară existența unor specificații sub forma unor **standarde deschise** referitor la modul de realizare a comenzii. Practic este necesară **standardizarea modului de transmitere a comenzilor**, prin rețeaua informațională a Hipersistemului CIM, respectiv **standardizarea comenzilor** utilizate. În cadrul acestui paragraf vor fi prezentate principalele probleme și standarde care au stat la baza realizării primelor structuri deschise de comandă în Hipersisteme CIM.

3.3.1. Care sunt problemele principale ale transmiterii comenzilor în Hipersisteme CIM?

Pentru realizarea coordonării activităților de fabricație este necesară transmiterea comenzilor prin rețeaua informațională a Hipersistemului CIM. Există mai multe standarde deschise care reglementează modul de transmitere a comenzilor sub forma unor mesaje, într-un mediu de rețea LAN. Prin aceste standarde se încearcă soluționarea următoarelor două probleme majore:

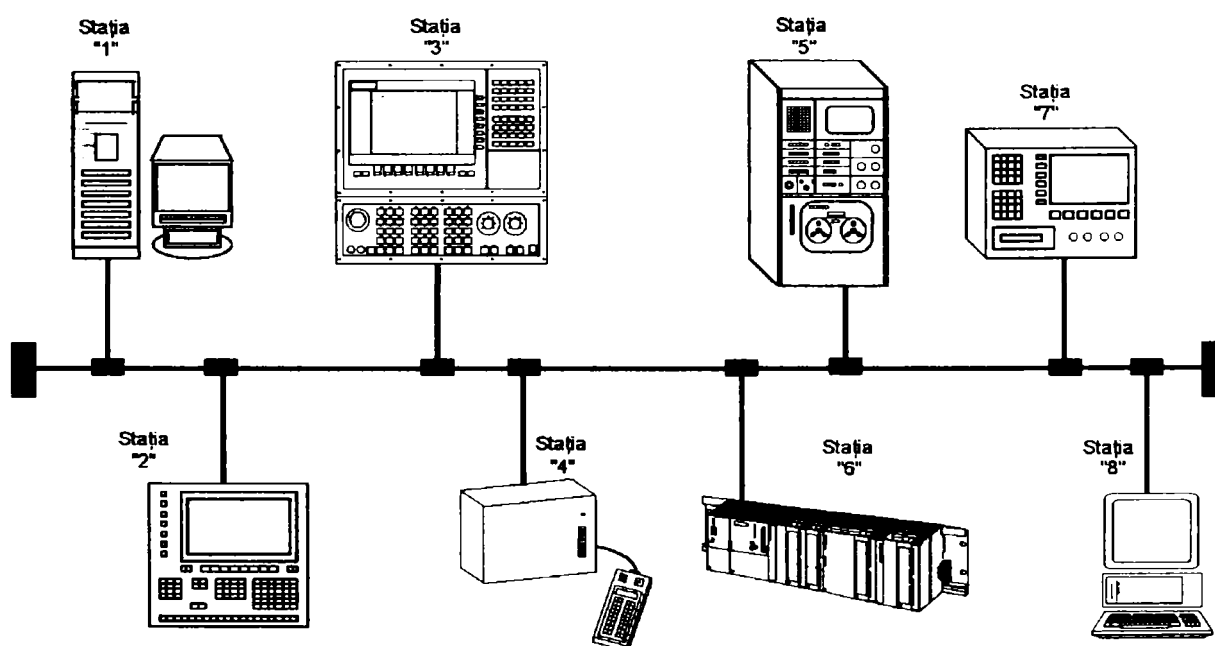


Fig.3.7. Exemplu de rețea LAN eterogenă din cadrul unui Hipersistem CIM

1. **Prima problemă** derivă din faptul că rețeaua informațională a Hipersistemului CIM este o **rețea eterogenă**, adică este formată din stații de tipuri și respectiv de proveniență diferită, fig.3.7. Dacă rețeaua ar conține numai stații identice, adică ar fi o rețea omogenă, nu ar fi o problemă realizarea unei comunicații între ele, ceea ce oricum este garantată de fiecare producător în parte pentru produsele lor proprii. În cazul rețelelor eterogene însă, trebuie rezolvată **problema transmiterii mesajelor între stații de tipuri și de proveniență diferită**.
2. **A doua problemă** derivă din faptul că, în cadrul unui Hipersistem CIM, fiecare echipament de comandă, este conectat la mediul fizic de transmisie (materializat prin cabluri coaxiale, cabluri cu fibre torsadate sau cabluri cu fibre optice), care reprezintă un **canal comun de comunicație**.

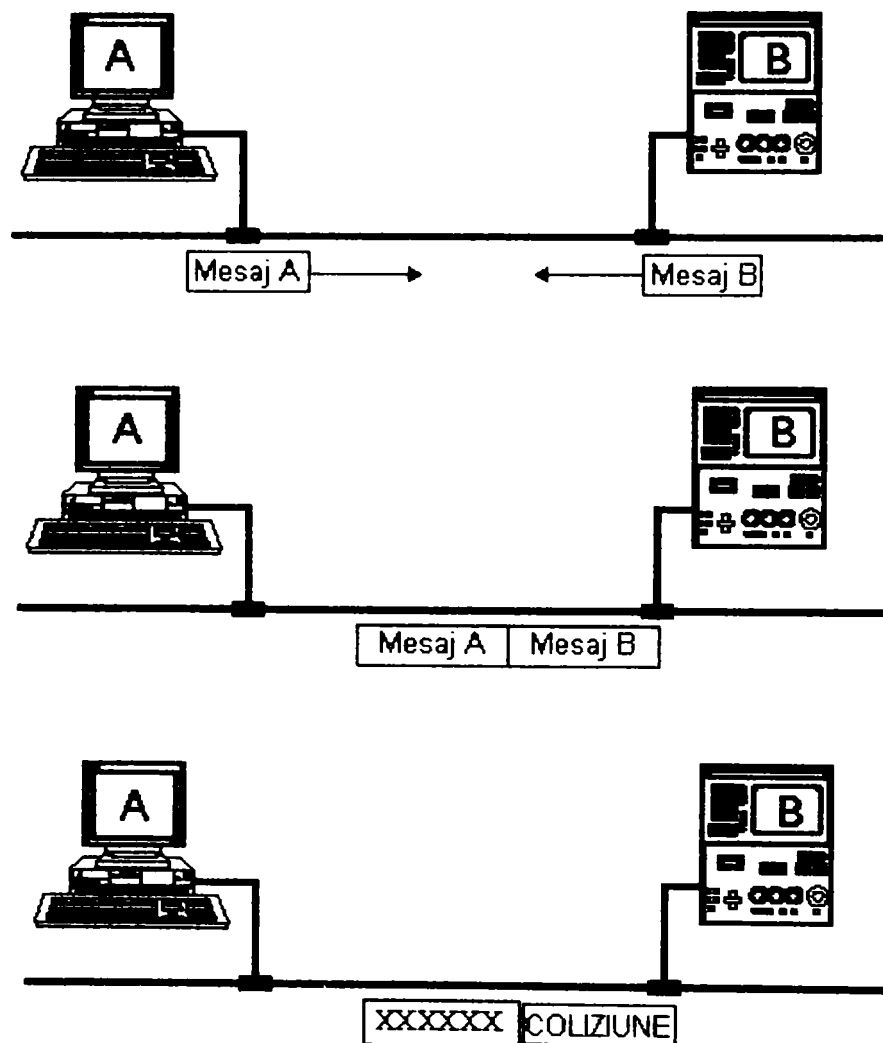


Fig.3.8. Ilustrarea cazului când două stații A, și B transmit mesaje simultan: apare o coliziune a datelor ce a ce conduce la distrugerea lor

Problema este că prin acest canal comun de comunicație, într-un anumit moment dat, numai o singură stație poate să transmită mesaje către alte stații. În acest timp celelalte stații trebuie să fie pe post de "recepție", pentru că în caz contrar apare o **coliziune** a mesajelor, care astfel sunt "distruse" și ca urmare rețeaua se blochează, (fig.3.8.). În consecință trebuie rezolvată **problema accesului la mediu** (accesul la canalul de comunicație), **a stațiilor din cadrul rețelei**.

3.3.2. Elaborarea unui model standard de transmitere a comenzilor între două structuri deschise - Modelul de referință OSI

Modelul de referință OSI a fost primul model standard prin care sa încercat soluționarea globală a problemei de transmitere a mesajelor informaționale, într-un mediu LAN eterogen. A fost elaborat în 1978 de către **Organizația Internațională de Standardizare** (International Organization for Standardization - ISO) sub denumirea de **Open Systems Interconnection (OSI) Reference Model** (Model de Referință pentru Interconectarea Sistemelor Deschise, sau simplu: **Model de Referință OSI**), [51]. Acest model a fost conceput sub forma unui standard care să faciliteze schimbul de informații între diverse sisteme deschise, [72].

Prezentarea modelului OSI

Pentru ilustrarea modului de rezolvare a problemei enunțate, propus în cadrul modelului, să considerăm două stații: 1 și 2 din cadrul Hipersistemului CIM, (fig.3.9.). De asemenea să considerăm că cele două stații sunt de tipuri respectiv de proveniență diferită (de exemplu un calculator pe rol de controler de celulă și un echipament de comandă numerică), respectiv sunt interconectate prin intermediul unui mediu fizic de transmisie (de exemplu cablu coaxial).

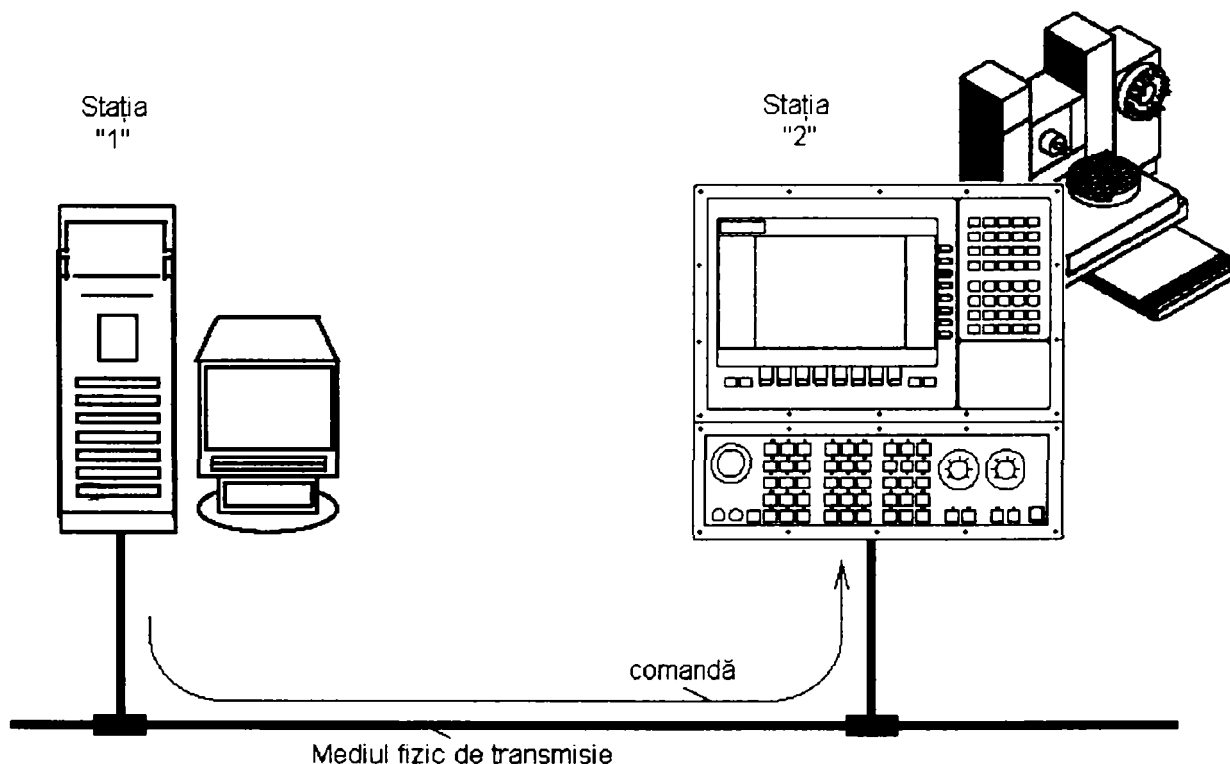


Fig.3.9. Transmiterea comenzilor între două stații

Pe stația 1 este rulat programul de aplicație A iar pe stația 2 programul de aplicație B. Schimburile de informații între cele două stații, sunt materializate prin transmiterea unor comenzi, sub formă de mesaje, de la programul de aplicație A către programul de aplicație B.

Pentru realizarea acestui schimb de informații, este necesar ca cele două programe de aplicație (adică atât cel care transmite cât și cel care recepționează) să realizeze codificarea respectiv decodificarea mesajelor care se transmit, sub formă digitală, după **reguli și procedee (soft și hard) identice**, adică să utilizeze același **protocol de comunicație**, (fig.3.10.).

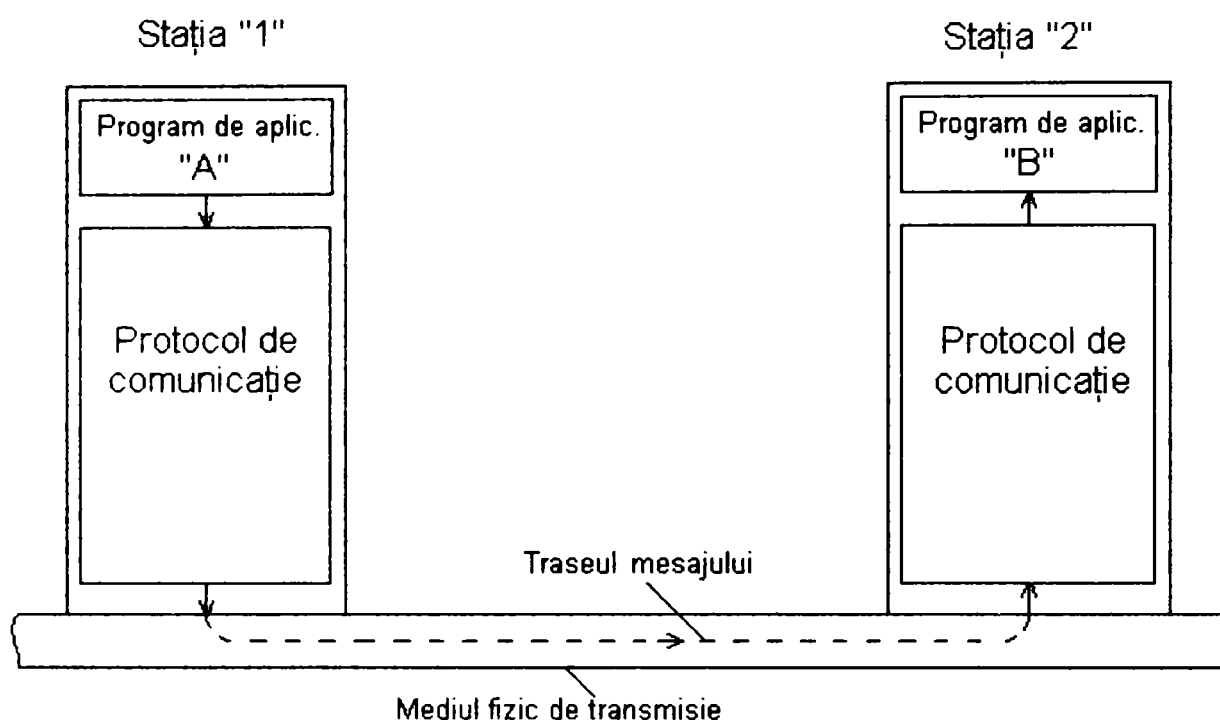


Fig.3.10. Transmiterea mesajelor între două programe de aplicație

În consecință, problema transmiterii mesajelor între două stații interconectate, se poate soluționa prin adoptarea și implementarea în cadrul fiecărei stații (la nivel hard și soft) a unui protocol de comunicație comun (identic).

Modelul de referință OSI a divizat problema de transmitere a mesajelor, în 7 probleme mai mici, care pot fi "gestionate" astfel mai ușor, [22]. Fiecare din cele 7 probleme se rezolvă prin intermediul a câte unui **layer** (nivel) al modelului la care corespunde câte un nivel al protocolului de comunicație (fig.3.11.).

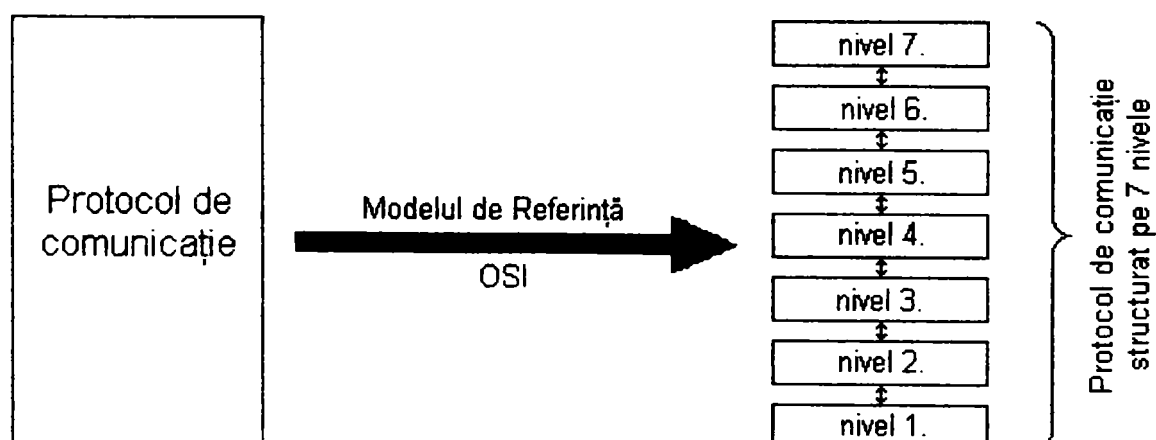


Fig.3.11. Structurarea protocolului de comunicație pe nivele

Modelul este astfel conceput încât fiecare nivel (al protocolului) pregătește și transmite nivelului următor anumite elemente, sub forma unor *servicii*. Astfel la rândul său fiecare nivel, utilizând serviciile oferite de nivelul imediat inferior, rezolvă problemele (sarcinile) care îi revin din transmiterea datelor corespunzătoare mesajului, iar "rezultatele" pune la dispoziția nivelului imediat superior.

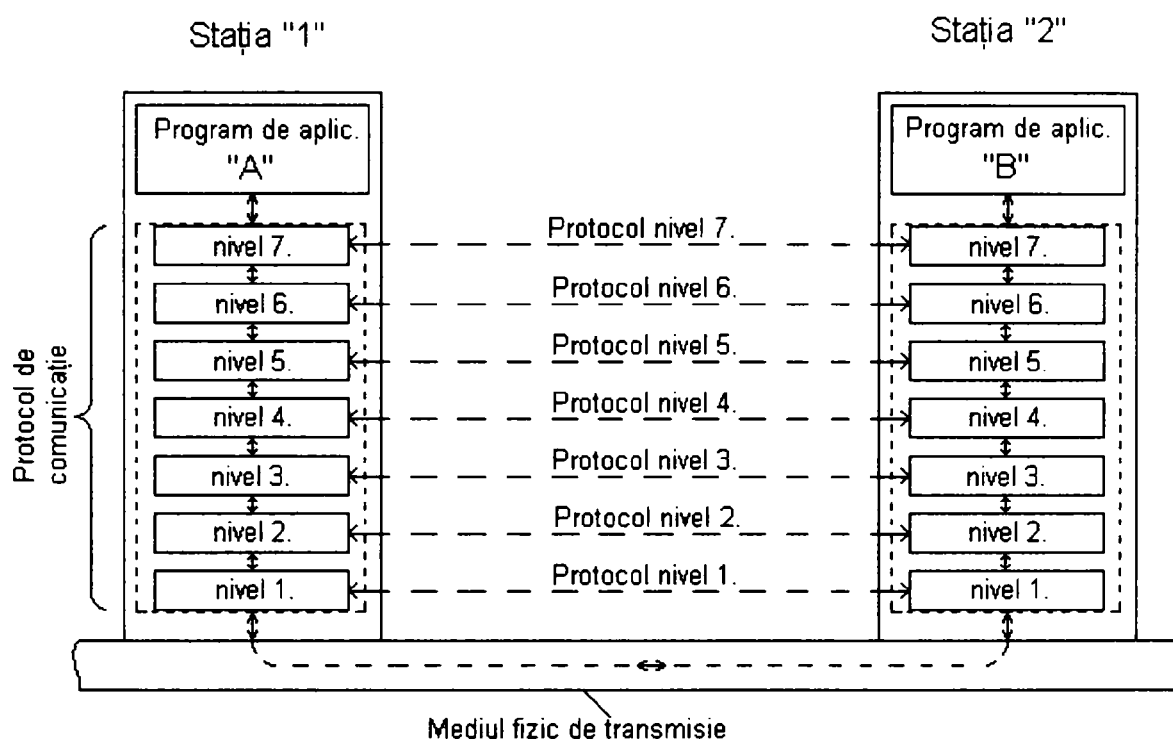


Fig.3.12. Transmiterea datelor între nivele, [51]

D.p.d.v. al comunicației dintre cele două stații (sisteme) interconectate, totul funcționează (în mod virtual) ca și cum fiecare nivel din cadrul stației 1 ar comunica, pe orizontală, direct cu nivelul identic din cadrul stației 2 (fig.3.12.). Și aceasta, datorită faptului că nivelele identice din cadrul celor două stații, conform modelului, utilizează același set de reguli și procedee referitoare la codificarea/decodificarea mesajului adică ele utilizează aceeași parte a protocolului de comunicație.

În realitate, transmiterea datelor corespunzătoare mesajului nu se realizează pe orizontală ci pe verticală. Adică datele care sunt transmise, de exemplu, de către un nivel al stației "1" către nivelul identic din cadrul stației "2", străbat întâi toate nivelele inferioare nivelului "expeditor" din cadrul stației "1" (de sus în jos), apoi sub formă de semnale electronice sunt transmise prin mediul fizic până la stația 2, unde străbat toate nivelele ale acestei stații de jos în sus, până când ajung la nivelul destinație. În mod asemănător se realizează transmiterea datelor, între cele două programe de aplicație, adică datele corespunzătoare mesajului transmis de programul "A" către programul "B" parcurg toate nivelele ale stației "1", de sus în jos, respectiv cele ale stației "2" de jos în sus. Pe baza modelului prezentat, este posibilă și comunicația inversă între stații, datele parcurgând invers traseul descris anterior.

În fig.3.13. s-a reprezentat schematic modelul OSI de transmitere a mesajelor între două programe de aplicații ale două stații interconectate. De asemenea au fost reprezentate denumirile standard ale celor 7 nivele (neglijându-se însă reprezentarea interfețelor dintre nivele).

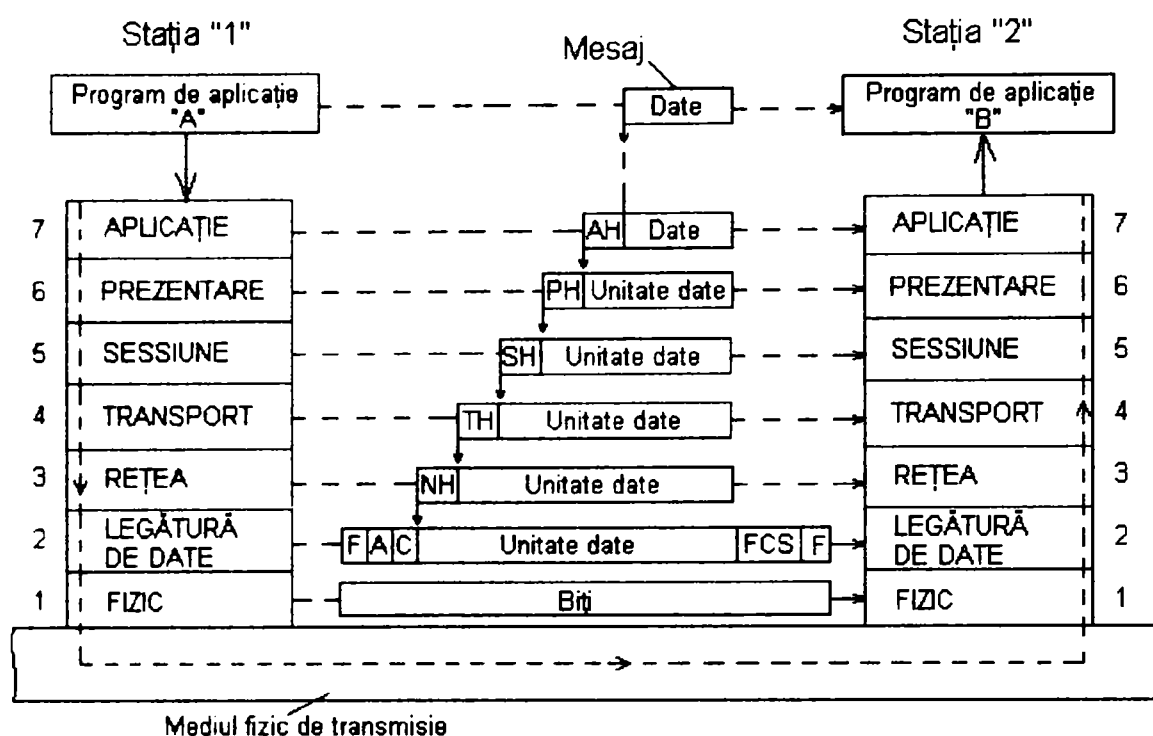


Fig.3.13. Modelul OSI de transmitere a unui mesaj între două stații, [20, 51]

Când programul de aplicație "A", de pe stația "1", are un mesaj de transmis programului de aplicație "B", de pe stația "2", va transfera acesta sub forma unor "date" nivelului de aplicație. Acest nivel îi adaugă acestor date un antet numit Application Header (AH). Informațiile din acest antet vor fi utilizate de nivelul aplicație din stația "2" pentru a interpreta datele și a le putea transmite programului "B" sub formă de mesaj.

Acum AH+Date este transmis nivelului prezentare, care îi consideră ca o **unitate de date** (Data Unit) și le adaugă un antet numit Presentation Header (PH). Nivelul de prezentare transmite noua unitate de date rezultată nivelului inferior și tot se repetă această operație, până la nivelul 2, de legătură de date. Acesta adaugă unități de date primite, în general atât un antet (header) cât și un marcaj terminal (trailer). Această unitate a nivelului 2, numită **cadru de date**, este apoi transferată nivelului fizic. Acesta vede cadrul de date sub forma unui șir de biți (o succesiune de 1 și respectiv 0) pe care îl transformă în semnale electronice. Aceste semnale se propagă prin mediul fizic până la stația destinație. Aici este inițiată procedura inversă.

Nivelul fizic al stației destinație, transformă semnalele electrice înapoi într-un șir de biți, care apoi este transmis nivelului legăturii de date. În timp ce datele "urcă" prin nivele, fiecare nivel elimină antetul (și eventualul marcaj terminal) cel mai exterior, procesează informația de protocol conținută în el și pasează ceea ce a rămas nivelului superior. Procesul continuă până când datele sosesc la programul de aplicație "B" care le va interpreta datele tot sub forma unui mesaj.

Observație:

- La fiecare pas al procesului, un nivel poate fragmenta unitatea de date în mai multe părți (este stabilită lungimea maximă a unităților respectiv a cadrelor de date). Aceste unități de date trebuie apoi reasamblate de către nivelele pereche corespunzătoare, înainte de a fi pasate mai sus.

Modelul prezentat în fig.3.13. poate fi utilizat pentru transmiterea mesajelor între stațiile unei subrețele LAN. În cadrul acestei subrețele fiecare stație trebuie să aibă, la fiecare nivel OSI, protocoale identice.

În cazul în care mesajele sunt destinate unei stații care se află într-o altă subrețea, transmiterea lor se realizează prin intermediul unor **echipamente de conectare**. Aceste echipamente, în cazul interconectării subrețelelor cu protocoale diferite, sunt pe rol de traducător (al unui protocol în celălalt și invers). Pe de altă parte au implementate numai acele nivele OSI (începând cu cele inferioare) care utilizează protocoale diferite în cadrul celor două rețele interconectate.

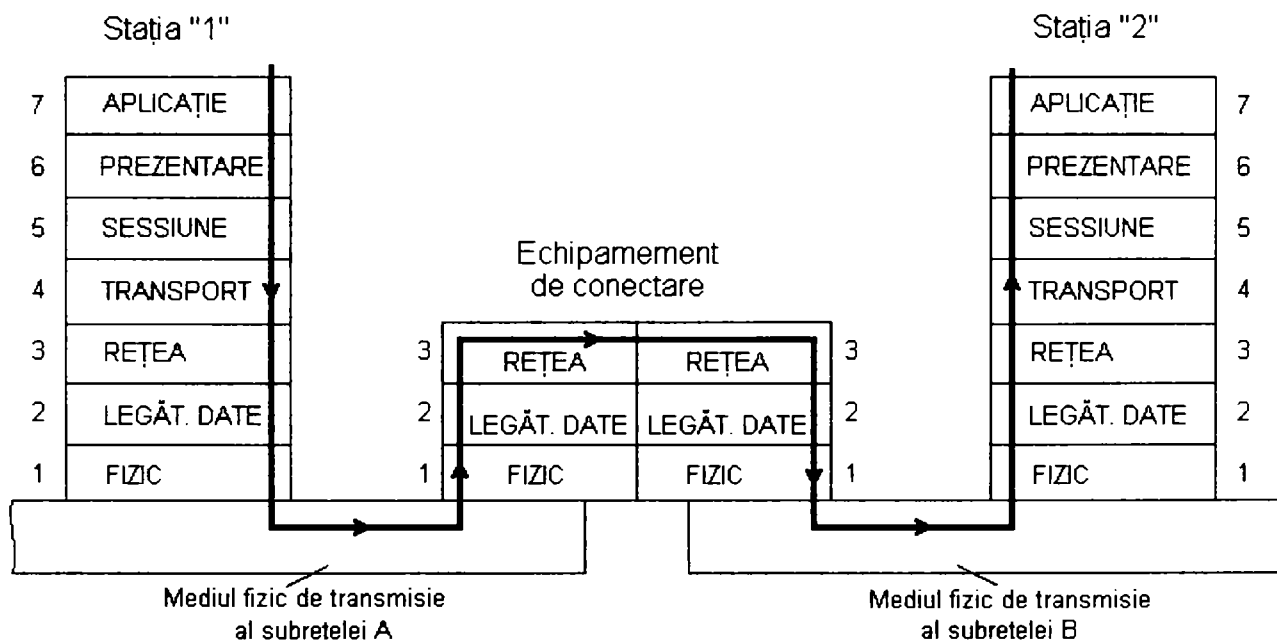


Fig.3.14. Modelul OSI de interconectare a două subrețele la nivelul 3, [51]

În fig.3.14 s-a prezentat modelul OSI de interconectare a două subrețele A și B la nivelul 3 printr-un echipament de conectare. Un astfel de echipament este numit: ROUTER. Interconectarea a două subrețele la nivelul 2 OSI se realizează cu un echipament numit PUNTE (Bridge) iar la nivelul 7 se realizează cu un echipament denumit POARTĂ (Gateway).

Funcțiile principale ale nivelelor OSI

În paragraful anterior s-au prezentat denumirile standard ale fiecărui nivel OSI. În continuare se prezintă funcțiile principale ale acestora.

<ul style="list-style-type: none"> • Nivel 7: Aplicație (APPLICATION) 	<p>Conectează programul de aplicație la procesul de comunicație și determină care funcții sunt necesare.</p>
<ul style="list-style-type: none"> • Nivel 6: Prezentare (PRESENTATION) 	<p>Se ocupă cu furnizarea datelor în format standard și cu conversia din forma lor originală într-o formă adecvată aplicației.</p>

• Nivel 5: Sesiune (SESSION)	Gestionează legăturile dintre programele de aplicații (conectarea, întreținerea și deconectarea legăturilor).
• Nivel 4: Transport (TRANSPORT)	Se ocupă cu gestiunea transportului de date.
• Nivel 3: Rețea (NETWORK)	Stabilește itinerarul pe care trebuie să parcurgă datele până la destinație, în cadrul unei rețele formate din mai multe subrețele.
• Nivel 2: Legătură de date (DATA LINK)	Se ocupă cu detecția și corecția erorilor apărute la transmisia prin mediul fizic, respectiv controlează accesul la mediul fizic.
• Nivel 1: Fizic (PHYSICAL)	Se ocupă cu codificarea și transmisia fizică a datelor. Realizează și controlul vitezei de transmisie.

Tab.3.1. Funcțiile principale ale nivelelor OSI, [18]

Pentru înțelegerea funcțiilor principale ale fiecărui nivel în transmiterea informației putem să facem o analogie cu transmiterea unei comenzi verbale prin telefon. Această analogie este bazată pe un material Siemens publicat în broșura lor "Comunicațiile ce aduc pacea în automatizări", [93]:

1. **Nivelul fizic:** Este ridicat receptorul și telefonul este conectat la "rețea". Se aude tonul.
2. **Nivelul legătură de date:** Controlul și detecția erorilor - este o linie bună fără zgomote.
3. **Nivelul rețea:** Este format numărul; un 9 pentru linia externă și apoi numărul propriu-zis. Telefonul sună la celălalt capăt .
4. **Nivelul transport:** Este ridicat receptorul la capătul celălalt - "Suntem ACME Products, puteți aștepta puțin, mă ocup de un alt apel. OK dați-i drumul. Nu am înțeles bine, puteți repeta vă rog?"
5. **Nivelul sesiune:** "Aici fabrica de Adezivi Aphrodite, am o comandă verbală pentru dvs., numărul CAP4057, contul meu este 7322D". Receptorul notează aceste detalii pentru situația în care legătura se întrerupe prematur.
6. **Nivel prezentare:** "Urmează numere de comenzi din catalogul pe Ianuarie 1993"
7. **Nivel aplicație:** - "Am mare nevoie de 100 bucăți din 302-706 și 50 bucăți din 209-417, trimise prin poștă în după-amiaza asta. Cost total 147,20 lire, veți primi factura".

Observații:

- În fiecare etapă, nivelele inferioare pot interacționa. Un zgomot pe linii de exemplu, va determina nivelul transport să solicite repetarea ultimului mesaj, [93].

3.3.3. Elaborarea unor metode standard de acces la mediul de transmitere a comenzilor

3.3.3.1. Metoda de acces la magistrală cu jeton (Token Bus)

Pentru rezolvarea problemei nr.2., adică a accesului stațiilor la canalul comun de comunicație, au fost elaborate mai multe standarde. Unul dintre ele este cel care se referă la subrețele de tip magistrală și promovează o metoda de acces cu jeton (IEEE 802.4).

Jetonul (numit în limba engleză "token") este un grup de biți și reprezintă pentru stația care îl posedă o "**permisiune exclusivă de transmisie mesaje**". Având în vedere faptul că, în cadrul unui segment de rețea, există numai un singur jeton și acesta poate să fie, la un anumit moment dat, în posesia doar a unei singure stații din cadrul segmentului, rezultă că totdeauna va exista maxim o stație, care va transmite mesaje către alte stații.

Datorită faptului că trebuie permis și la alte stații să transmită mesaje, jetonul trebuie să între, pe rând, în posesia fiecărei stații din cadrul segmentului. "**Pasarea**" jetonului la următoarea stație, se realizează prin intermediul unui procedeu, care ține cont de o ordine logică, dată de **numerele de adresă** ale fiecărei stații în parte (de exemplu: 8-7-6-5-4-3-2-1) și nu de ordinea fizică (8-7-5-6-4-3-2-1) în care sunt conectate stațiile la magistrală (fig.3.14). Jetonul este întotdeauna pasat (transmis) stației care are adresa (de rețea) următoare cea mai mică. După terminarea unui ciclu complet, jetonul (ajuns în posesia stației cu adresă de rețea cea mai mică) este pasat stației care are adresa de rețea cea mai mare și astfel ciclul începe din nou.

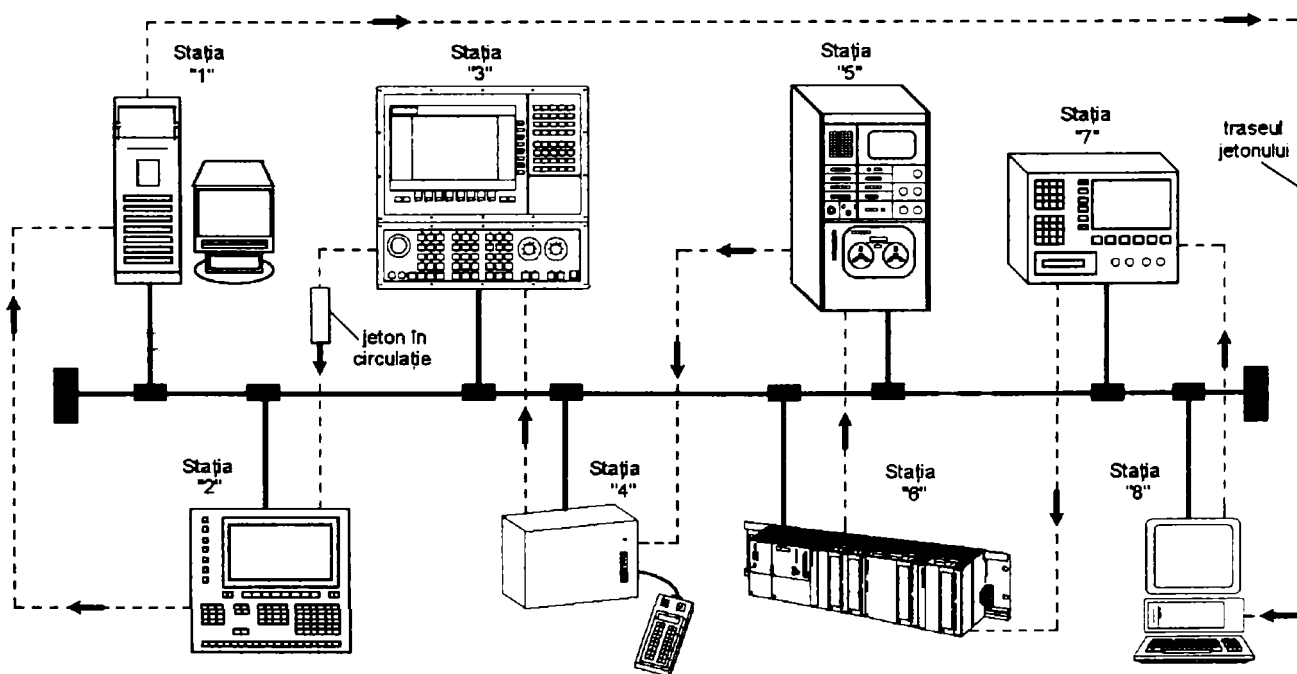


Fig.3.14. Ilustrarea circulației jetonului în cadrul segmentului de rețea

Când jetonul este recepționat de o stație și aceasta nu are nimic de transmis, atunci stația îl transmite mai departe următoarei stații conform procedurii descrise. Dacă jetonul este acceptat, el este retransmis mai departe după ce stația a terminat

transmisia mesajelor proprii. Jetonul trebuie livrat stației următoare într-un interval de timp prestabilit, astfel încât nici o stație nu poate monopoliza resursele rețelei. Astfel într-un timp deterministic fiecare stație intră, pe rând, în posesia jetonului și astfel poate să transmită mesaje.

Deși rețeaua este d.p.d.v. fizic de tip magistrală, jetonul descrie totdeauna traseul unui inel logic (fig.3.15.).

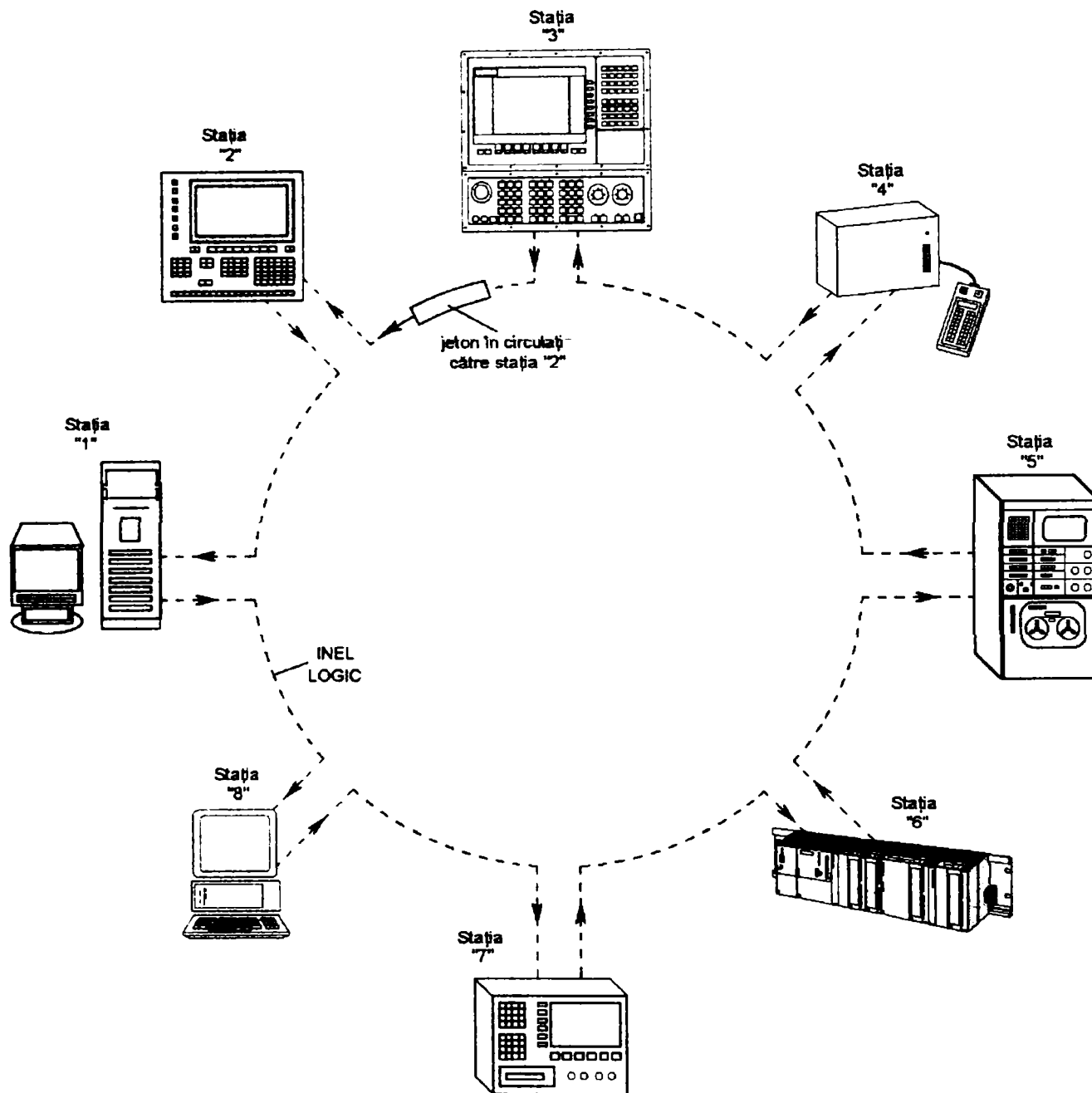


Fig.3.15. Traseul jetonului este un inel logic

Fiecare stație cunoaște adresa **predecesorului**, adică a stației de la care primește și a **succesorului**, adică a stației căreia îi transmite jetonul, [18]. Când segmentul de rețea este inițializat sau când este restabilit după un defect, fiecare stație trebuie să-și stabilească predecesorul și succesorul. Pe măsură ce sunt adăugate sau eliminate stații din inel, stațiile își actualizează valorile de predecesor și succesor. Pentru a se adăuga o stație în inel, o stație ce a recepționat jetonul transmite un cadru de control, denumit "**solicit-succesor**", ce conține adresa proprie a stației și

adresa succesului. Orice stație din segment ce are o adresă ce cade în gama acestor două adrese poate solicita să fie adăugată în inel.

Procedurile de menținere a inelului sunt proiectate astfel încât se pot adăuga stații doar în momentele în care traficul în rețea este relativ scăzut. După ce stația ce deține jetonul a transmis cadrul solicit-succesor, va aștepta o perioadă de timp denumită **fereastră de răspuns** pentru a vedea dacă răspunde o stație. Dacă nu este sesizat nici un răspuns, se presupune că nici o nouă stație nu dorește să fie adăugată în inel.

Dacă însă apare un răspuns, o nouă stație este adăugată în inel, atribuindu-se o adresă în gama specificată de cadrul solicit-succesor. Dacă răspund mai multe stații în același timp apare o coliziune, pentru rezolvarea căreia vor fi utilizate proceduri de rezolvare a concurenței.

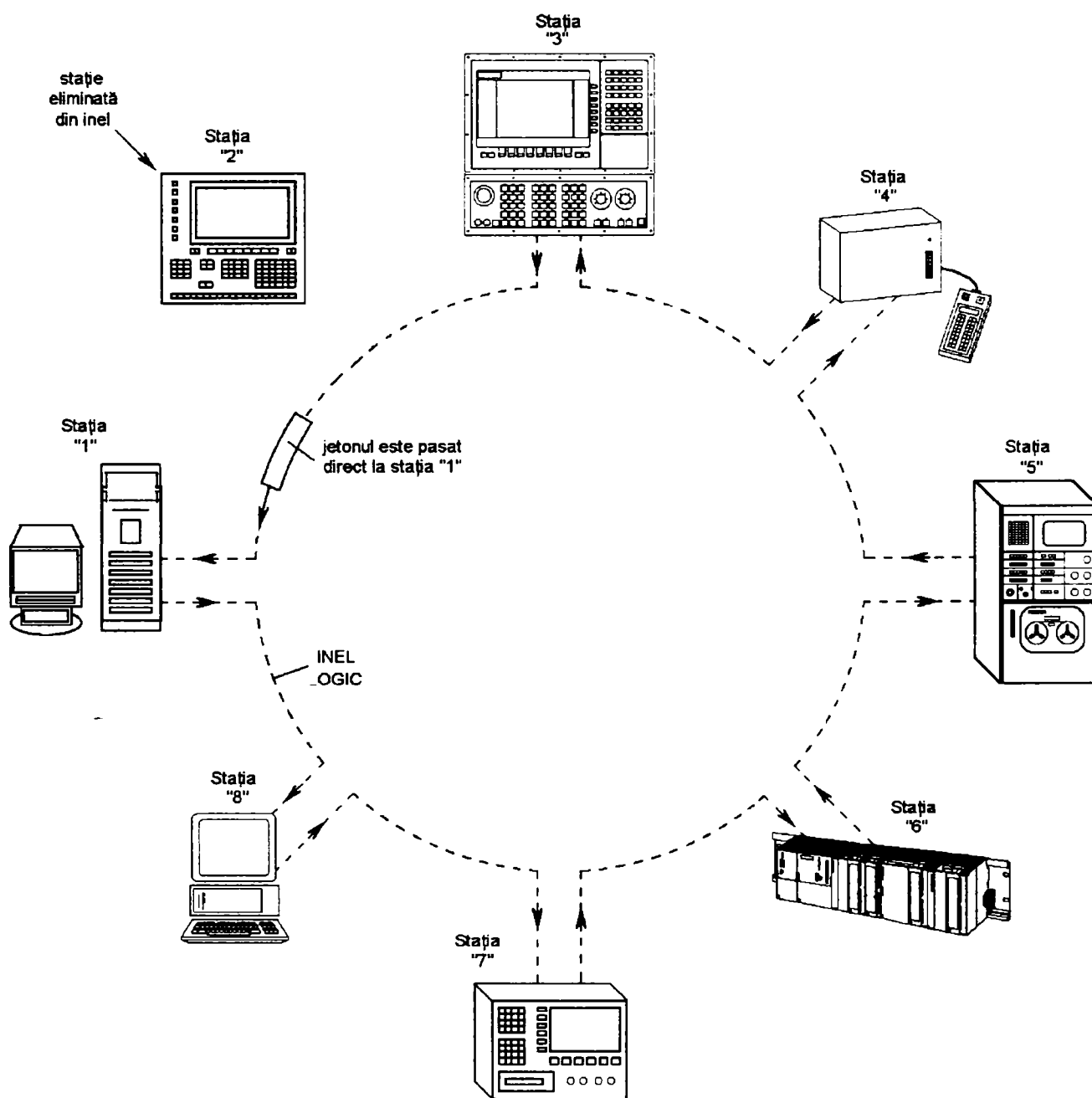


Fig.3.16. Eliminarea stației "2" din inel prin scurtcircuitarea traseului urmat de jeton

Procesul eliminării unei stații este mai simplu decât cel de adăugare. Atunci când o stație dorește să iasă din inel va aștepta până ce va recepționa jetonul. Va transmite un cadru "**set-succesor**" către predecesor, notificând schimbarea adresei succesorului. De acum în continuare predecesorul va pasa totdeauna jetonul direct către succesorul stației eliminate, scurtcircuitând stația eliminată, [18].

În fig.3.16. s-a reprezentat cazul eliminării stației "2" din inel. Ca urmare a eliminării, stația "3" va avea noul succesor pe stația "1" în loc de stația "2" și ca atare jetonul va fi pasat totdeauna direct la stația "1" (urmărind ordinea: 8-7-6-5-4-3-1-8). Atunci când stația "2" va dori reintrarea în inel, va trebui să inițieze procedura de adăugare a unei stații în inel.

3.3.3.2. Metoda de acces cu sesizarea purtătoarei (Ethernet)

La această metodă de acces la magistrală, termenul de **sesizare de purtătoare** se referă la faptul că o stație **ascultă înainte de a transmite**. Adică se utilizează o metoda prin care o stație care dorește să emite un mesaj ascultă în "rețea" să vadă dacă este sau nu utilizată de altcineva. Dacă este atunci stația așteaptă. Dacă mediul fizic nu este ocupat, stația emite mesajul (prin aceasta blocând toate celelalte stații până când se termină mesajul). Această metoda se numește **Acces Multiplu cu Sesizare de Purtătoare** (Carrier Sense Multiple Acces - CSMA).

Totuși pot apare situații neplăcute, atunci când 2 sau mai multe stații încep simultan să emite mesaje, provocându-se o coliziune a mesajelor. Această situație poate fi ușor detectată, ambele stații oprindu-se și așteptând un timp oarecare înainte de a încerca să retransmită mesajul. Este utilizat un timp aleator de oprire a celor 2 stații pentru a nu se repeta coliziunea. Aceasta se numește **Acces Multiplu cu Sesizare de Purtătoare și Detecția Coliziunii** (Carrier Sense Multiple Acces with Collision Detection - CSMA/CD), [49].

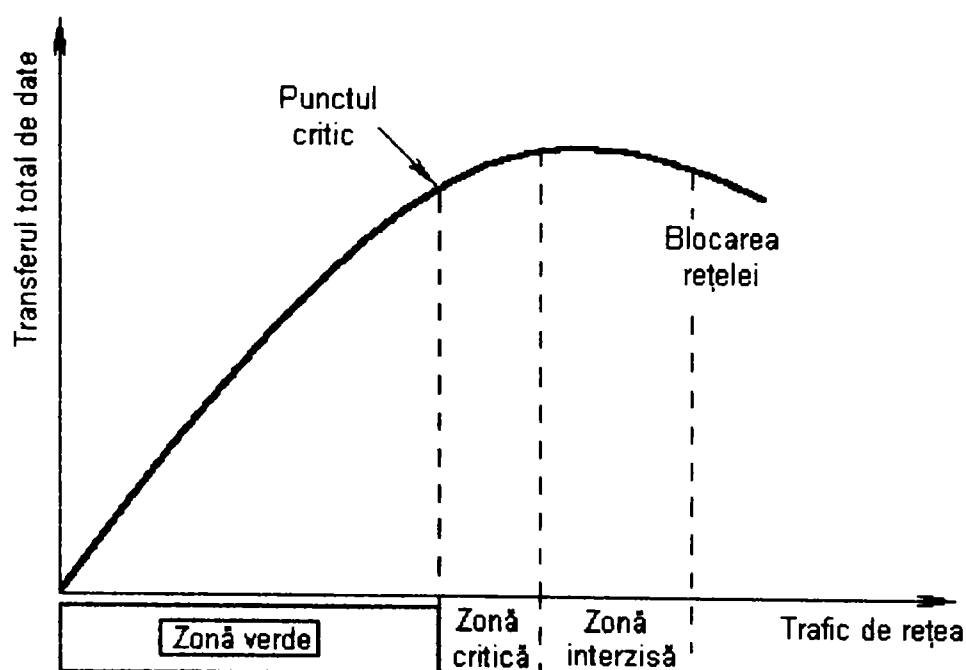


Fig.3.17. Relația între mărimea traficului și transferul efectiv de date, [96]

Un avantaj major al metodei CSMA/CD este acela că transferul datelor se face foarte rapid atâta timp cât traficul nu este foarte mare, deoarece o stație poate transmite oricând mediul fizic este liber. La încărcări mari ale rețelei (adică în cazul traficului intens) numărul de coliziuni crește și în consecință intervalul de timp necesar pentru tratarea coliziunilor și retransmisie, face ca performanța să se degradeze. Chiar poate să se ajungă într-o situație, (fig.3.17.), în care numărul mare al coliziunilor împiedică complet transferul de mesaje între stații (blocarea rețelei).

Totuși, coliziunile nu produc o problemă de trafic până la încărcări mari ale mediului. Până la 90% din capacitatea canalului poate fi utilizată tehnica CSMA/CD. Cu toate acestea, în implementările practice se încearcă evitarea utilizării acestei metode la încărcări mai mari de 40-50%.

O altă formă de control al accesului la mediu, înrudită cu CSMA/CD este denumită **Acces Multiplu cu Sesizare de Purtătoare și Evitarea Coliziunii** (Carrier Sense Multiple Acces with Collision Avoidance - CSMA/CA). În cazul acestei metode fiecare stație ascultă mediul cât timp fiecare transmisie este în curs. După ce transmisia se termină, fiecare stație așteaptă un anumit interval de timp, bazat pe poziția sa relativă dintr-o listă logică a stațiilor. Dacă nici o altă stație nu a început transmisia atunci când i-a venit rândul, ea poate începe să transmită.

Observații:

1. Metoda de acces la magistrală de tip CSMA/CD este utilizată în așa numitele "rețele Ethernet".
2. Metodele de acces la canalul comun de comunicație (la mediul fizic) sunt protocoale ale nivelului 2 OSI (Legătură de Date), [18].

3.3.4.Elaborarea unor metode standard de transmitere referitoare la mediul fizic

Pentru transmiterea în mediul fizic a datelor aferente comenzilor emise în cadrul structurilor de fabricație automatizate se utilizează 3 metode standard:

1. în bandă de bază (baseband),
2. cu purtătoare (carrierband),
3. în bandă largă (broadband).

La metoda de **transmisie în bandă de bază (baseband)** semnalul informațional (adică datele digitale codificate în binar), este transmis (sub forma unui șir de cifre de 1 și respectiv 0) prin mediul fizic, direct (adică fără modulație), sub forma unor impulsuri (electrice) digitale. Transmisia datelor se realizează la frecvențele lor originale (adică de bază). Prin acest mod întregul mediu fizic formează un singur "canal de comunicație" și în consecință pot fi transmise date, la anumit moment dat, numai de către o singură stație din cadrul segmentului de rețea.

La metoda de **transmisie cu purtătoare (carrierband)**, transmisia datelor digitale (adică a semnalului informațional) se realizează cu ajutorul unui semnal analogic numit *semnal purtător*, peste care este suprapus (prin modulare) semnalul infor-

mațional. Modularea semnalului informațional, pe un semnal purtător, poate fi realizat în general prin trei modalități, (fig.3.18.): prin *modulație în amplitudine (ASK)*, prin *modulație în frecvență (FSK)* și prin *modulație în fază (PSK)*, [66, 93].

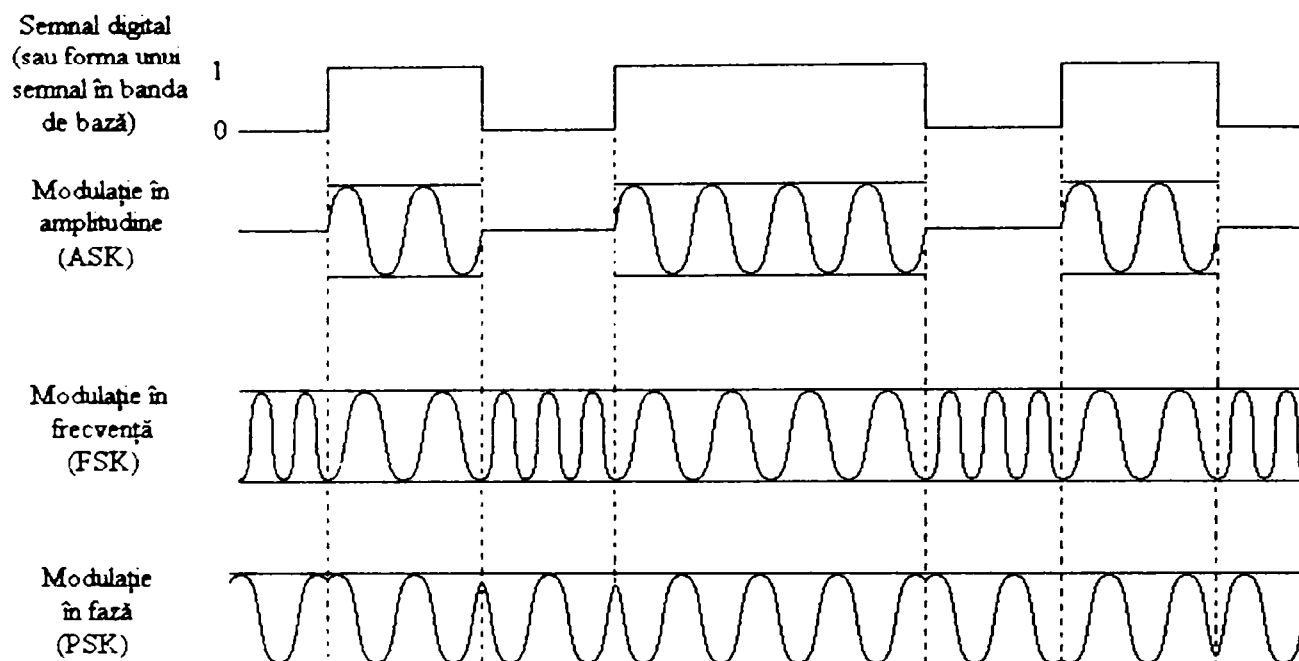


Fig.3.18. Modularea semnalelor digitale

Prin modulare, datele digitale sunt transformate practic în semnale analogice. În această formă, prin posibilitatea utilizării amplificatoarelor analogice, datele sunt mult mai imune la perturbații decât în cazul transmisiei în banda de bază (fără purtătoare). La recepție însă, trebuie să aibă loc un proces de demodulare prin care semnalul informațional este separat de semnalul purtător și ca urmare este reobținută forma digitală a datelor.

Indiferent de tipul modulării toate semnalele de informație sunt modulate pe un singur "purtător". Astfel pot fi transmise date, la un anumit moment dat, numai de către o singură stație adică există de asemenea numai un singur "canal de comunicație".

La metoda de **transmisie în bandă largă (broadband)**, transmisia datelor digitale se realizează de asemenea cu ajutorul semnalelor purtătoare, însă pe baza unui procedeu există posibilitatea "divizării" singurului canal de comunicație din cadrul mediului fizic, în mai multe (sub-) canale distincte. Practic se realizează divizarea benzii de frecvență a mediului fizic într-un număr de benzi distincte numite "canale", (asemănător transmiterii prin cablu a mai multor canale de televiziune), [68].

Procedeu utilizat se numește *Multiplexare cu Divizare în Frecvență (Frequency Division Multiplexing – FDM)* prin care benzile de frecvență ale canalelor întâi sunt decalate cu valori diferite, după care sunt reunite, (fig.3.19.), [36].

În cadrul structurilor de automatizare a fabricației, metoda de transmisie în bandă largă permite transmiterea simultană a comenzilor de către multe stații, prin utilizarea mai multor canale (de comunicație) distincte.

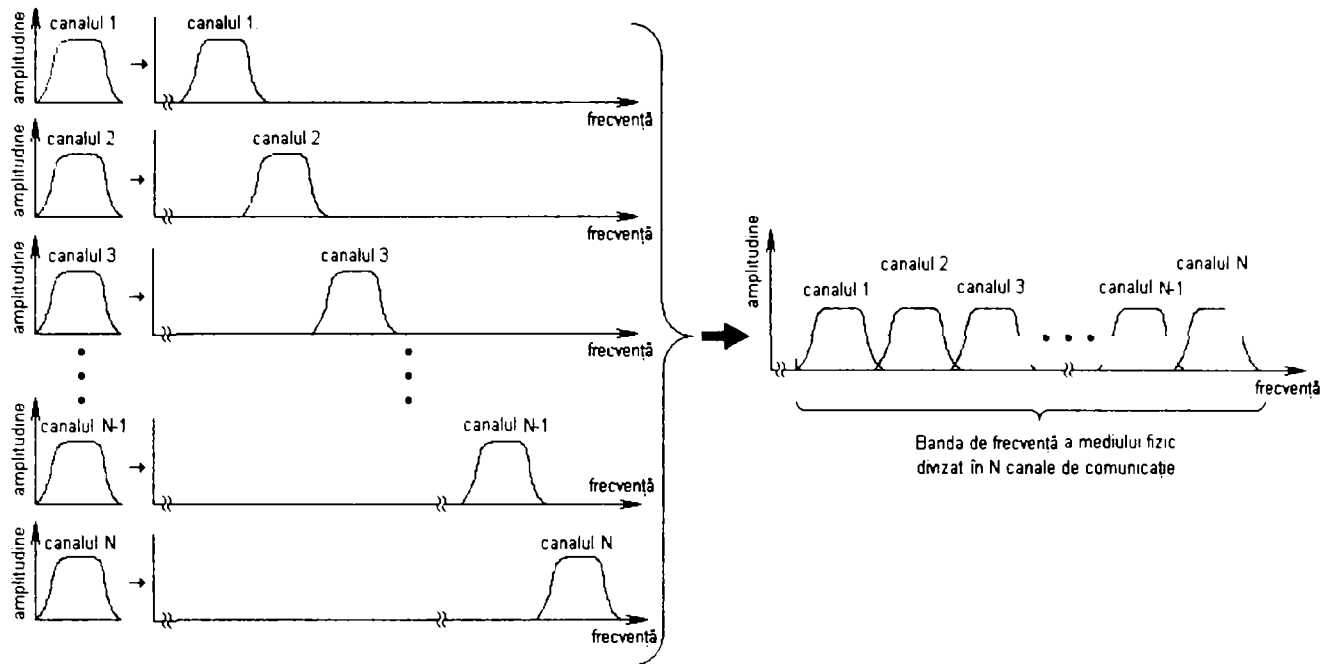


Fig.3.19. Principiul divizării benzii de frecvență a mediului fizic în "N" canale distincte

Transmisiile de tip multi-canal (bandă largă) sunt cele mai eficiente în transmiterea mesajelor aferente comenzilor, mai ales în cazul traficului intens. Din acest motiv, rețeaua coloană de fabrică din cadrul Hipersistemelor CIM este "în mod tradițional" de tip bandă largă, ce permite de exemplu o comunicație simultană, cu fiecare rețea de celulă, prin utilizarea unor canale distincte.

3.3.5. Apariția primelor structuri deschise destinate transmiterii comenzilor în Hipersisteme CIM. Apariția MAP-ului

Procedeul de elaborare a unor standarde deschise referitor la comanda la distanță a diferitelor echipamente de comandă, a fost inițiată prima oară de firma General Motors (GM), la începutul anilor '80, printr-un proiect de lungă durată.

Această necesitate de standardizare a apărut chiar la GM, datorită faptului că aceasta firmă avea în dotare între altele 20000 de echipamente programabile, 2000 de roboți respectiv 40000 de echipamente "inteligente", [101]. Există o previziune privind ridicarea acestor cifre de dotare la un nivel de 5 ori mai mare în următorii cinci ani. Cu ridicarea gradului de automatizare crește în mod incontestabil eficacitatea producției, dar posibilitățile vor putea fi exploatate mai bine doar dacă se integrează procesele de automatizare într-o bază de date comună. Fără o rezolvare în ansamblu a problemei va fi posibil un schimb de informații doar între 15% a proceselor, [28, 101]. Astfel există posibilitatea apariției unui număr mare (la GM peste 1000) de "insule de automatizare".

Observând lipsa standardelor interne din cadrul GM, conducerea a înființat încă în 1980 grupul **MAP Task Force**. Sarcina acestui grup a fost inventarierea tuturor modalităților de schimb de informații utilizate de diferiți producători de echipamente în cadrul automatizării fabricației. Grupul avea în componență reprezentanți din circa 15 secții ale firmei GM. Acest grup a lansat strategia de standardizare a schimburilor in-

formaționale în cadrul structurilor de fabricație automatizate, prin elaborarea unor seturi de specificații, cunoscute sub numele de **Protocol de Automatizare a Fabricației** (MAP - Manufacturing Automation Protocol). Acesta fiind destinat tuturor producătorilor de echipamente de comandă, a fost lansat în ideea că dacă în viitor toți fabricanții vor respecta aceste specificații atunci nu vor mai exista probleme de incompatibilitate informațională la integrarea lor în Hipersisteme de tip CIM.

Prima versiune (versiunea 1.0) a specificațiilor MAP a fost publicată (de General Motors) în 1982. Ca urmare a deficiențelor inerente care a caracterizat această prima versiune în 1985 au fost publicate versiunile îmbunătățite MAP2.0. respectiv MAP2.1., [73]. Versiunea actuală MAP3.0 a fost publicată în 1988 și completată cu câteva amendamente în 1993, [6].

Deși proiectul MAP este legat fără echivoc de numele firmei General Motors, au aderat la această idee de standardizare și o serie de alte firme. Acestea din urmă sunt în mare măsură fabricanți și furnizori de sisteme electronice ai căror cointeresare a fost cauzată de acea directiva a firmei GM care a prevăzut că din 1990 firma va achiziționa numai acele echipamente care sunt dotate cu interfețe realizate pe baza specificațiilor MAP (interfețe MAP).

Se poate considera că apariția și lansarea MAP-ului a fost inevitabilă. În 1980 existau pe piață mai multe tipuri de rețele de calculatoare dezvoltate pentru schimburi de date și informații. Dar, datorită faptului că aceste rețele au provenit din dezvoltări izolate, din cadrul unor firme distincte, interconectarea lor a fost total imposibilă. Trebuia așteptat până când o firmă **utilizatoare** de echipamente electronice, care era și într-o situație financiară extrem de bună, a recunoscut această necesitate de a se "**face ordine**" pe planul transmiterii informațiilor, [86].

Elaborarea MAP-ului (Protocol de Automatizare a Fabricației) de către firma General Motors, în scopul obținerii unui mod de transmitere standard a comenzilor în cadrul structurilor de fabricație automată, s-a realizat prin stabilirea unor specificații concrete, sub formă de standarde internaționale, pentru funcțiile și respectiv serviciile fiecărui nivel OSI în parte, [22]. Aceste funcții și servicii au fost stabilite în concordanță cu condițiile și cerințele transmiterii comenzilor în mediu industrial.

După posibilitățile de transmitere în timp real a comenzilor în cadrul structurilor MAP se deosebesc două arhitecturi: arhitectura Full-MAP și arhitectura Mini-MAP.

3.3.5.1. Arhitectura Full-MAP

Structurile cu arhitectura Full-MAP au implementate "complet" (full) toate cele 7 nivele OSI ale Protocolului de Automatizare a Fabricației (MAP). Atunci când se fac referiri generale la structuri MAP, practic se are în vedere această arhitectura Full-MAP, [29].

În fig.3.20. s-a reprezentat arhitectura generală Full-MAP. Reprezentarea ține cont de specificațiile MAP3.0, varianta din 1993. Această variantă aduce noutăți (în sensul completării), față de varianta MAP3.0 din 1988 (când a fost publicată prima

oară versiunea 3.0) și anume permite utilizarea în cadrul structurilor MAP a “metodei” de acces la mediu (magistrală) de tip CSMA/CD, alături de protocolul “tradițional” de acces cu jeton (Token Bus).

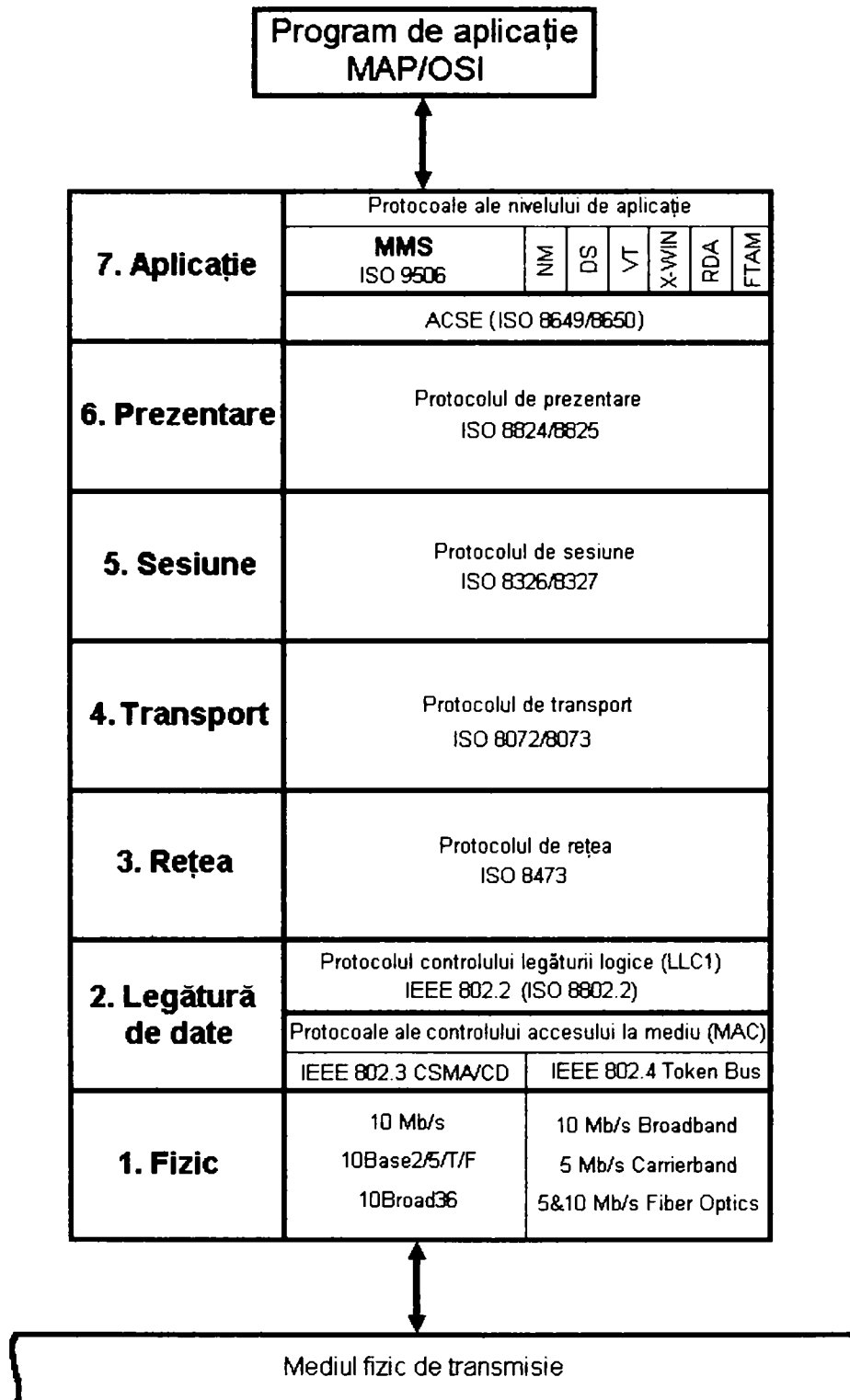


Fig.3.20. Arhitectura Full-MAP

În acest sens sunt acceptate specificațiile pentru transmisia prin cabluri coaxiale (10Base2, 10Base5, 10Broad36), pentru transmisia prin cabluri cu fire torsadate (10baseT) și respectiv pentru transmisia prin cabluri cu fibre optice (10BaseF), ca făcând parte integral din cadrul specificațiilor MAP, [31].

Desigur, au fost menținute specificațiile referitoare la mediul fizic, în cazul utilizării protocolului de acces la magistrală cu jeton (Token Bus) și anume: specificațiile referitoare la transmisia prin cabluri coaxiale în bandă largă (10Mb/s Broadband) respectiv cu purtătoare (5Mb/s Carrierband), precum și cele referitoare la transmisia prin cabluri cu fibre optice (5&10Mb/s Fiber Optic).

La nivelul de aplicație poate fi identificat un serviciu numit MMS. Acesta conform standardului ISO9506 oferă programului de aplicație un schimb de mesaje de fabricație (manufacturing messages) cu alte stații. Alte "servicii" oferite de nivelul de aplicație, ale arhitecturii Full-MAP sunt:

1. **FTAM (File Transfer, Acces, and Management)** ce permite transferul, accesul și gestionarea fișierelor (ISO 8571), [52].
2. **DS (Directory Services)** care furnizează informațiile despre obiectele (stațiile) din respectivul segment de rețea sau din alt segment cu care se interconectează, (ISO 9594), [61]. DS permite conversia numelui de rețea, furnizat de o cerere de serviciu, într-o adresă de rețea corespunzătoare.
3. **VT (Virtual Terminal)** ce permite programelor de aplicații să capete acces la terminale aflate la distanță, indiferent de varianta lor constructivă, (ISO 9040/9041), [56, 57].
4. **NM (Network Management)** oferă servicii de gestiune a schimburilor de date în vederea obținerii unor eficiențe maxime.
5. **RDA (Remote Database Acces)** oferă programelor de aplicații să capete acces la baze de date aflate la distanță, (ISO 9579), [60].
6. **X-Windows** oferă utilizatorilor o interfață pentru programarea aplicațiilor, independentă de echipament. În mod asemănător programelor Windows 3.1 respectiv 95, este bazat pe utilizarea ferestrelor ("windows").
7. **ACSE (Association Control Service Element)** oferă servicii privind stabilirea, controlul și terminarea unei "legături" (asociații) între două programe de aplicații. Reprezintă subnivelul inferior al nivelului de aplicație, [53, 54].

Observație:

- Conform specificațiilor MAP, varianta de bandă largă (broadband) a arhitecturii Full-MAP, este utilizată în general în cadrul rețelei coloană de fabrică (Factory Backbone Network) pentru realizarea comunicației simultane, cu mai multe rețele de celulă conectate la rețeaua coloană de fabrică.

Avantajul mare al acestei arhitecturi "complete" (Full-MAP) este faptul că, respectă întocmai prevederile Modelului de referință OSI, referitoare la interconectarea sistemelor deschise.

Ca **dezavantaj** se poate aminti faptul că nu sprijină utilizarea programelor de aplicație în timp real, care sunt strict necesare pentru comanda (în timp real) a celulelor de fabricație. Aceste programe, necesită un timp redus pentru transmiterea mesajelor de fabricație între stații. Însă, arhitectura Full-MAP nu poate garanta acest lucru, motivul fiind numărul prea mare de prelucrări la care sunt supuse datele transmise, la parcurgerea celor 7 nivele ale protocolului. Astfel se recomandă, pentru programe de comandă în timp real, *arhitectura Mini-MAP*.

3.3.5.2. Arhitectura Mini-MAP

Structurile cu arhitectură Mini-MAP utilizează un număr de numai 3 nivele: fizic, legătură de date și aplicație. Nivelele de rețea, transport, sesiune și prezentare sunt înlocuite cu o interfață soft, care asigură un transfer rapid și de foarte scurtă durată a datelor între nivelul de aplicație și nivelele inferioare (fizic și legătură de date). Astfel, această arhitectură sprijină utilizarea programelor de comandă în timp real.

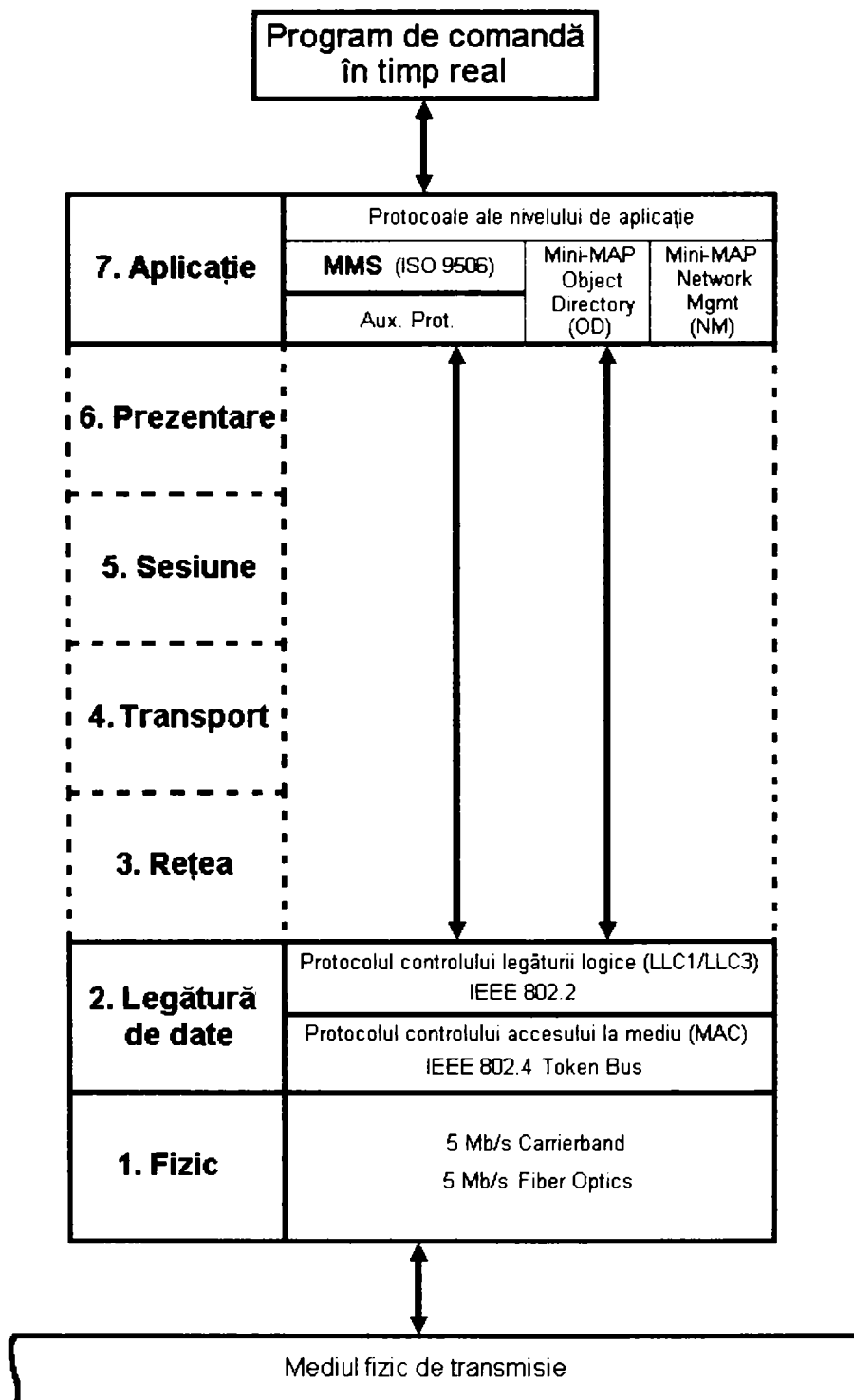


Fig.3.21. Arhitectura de rețea Mini-MAP

În fig.3.21. s-a reprezentat arhitectura generală de rețea de tip Mini-MAP, [91]. Se poate observa, pe lângă lipsa celor 4 nivele amintite, posibilitatea utilizării a nu-

mai unei singure metode, referitoare la controlul accesului la mediu și anume metodei care garantează acest acces și în cazul unui trafic intens (Token Bus). De asemenea, se poate observa că stațiile care au implementate această arhitectură, vor utiliza pentru comunicație, transmisia de tip carrierband, pe cablu coaxial sau pe cablu cu fibre optice. Față de arhitectura Full-MAP, subnivelul LLC al nivelului 2. de legătură de date oferă posibilitatea confirmării "imEDIATE" a datelor transmise (LLC3), ceea ce este foarte important pentru garantarea unui timp redus de transfer, [55].

Această arhitectură Mini-MAP, datorită performanțelor sale ridicate d.p.d.v. al comunicației în timp real, mai este numit și **arhitectură EPA (Enhanced Performance Architecture)**, [28].

Dezavantajul acestei arhitecturi, este faptul că nu respectă recomandările Modelului de referință OSI, referitoare la interconectarea sistemelor deschise, adică o stație care are implementată această arhitectură Mini-MAP nu poate comunica cu alte sisteme (stații) OSI. În consecință, în cadrul segmentului de rețea în care este prevăzută a fi utilizată (pentru scopuri de interfațare a programelor de comandă în timp real), fiecare stație trebuie să aibă implementată obligatoriu această arhitectură Mini-MAP.

Datorită acestor probleme, conectarea unei subrețele de celulă de tip Mini-MAP la rețeaua coloană de fabrică, necesită utilizarea unui echipament de tip Poartă (Gateway). Acest echipament, datorită faptului că realizează "compatibilizarea" celor două arhitecturi: Full-MAP și Mini-MAP (EPA) este numit "Poartă MAP/EPA". În unele cazuri aceste echipamente sunt și pe rol de controler de celulă, fig.3.22.

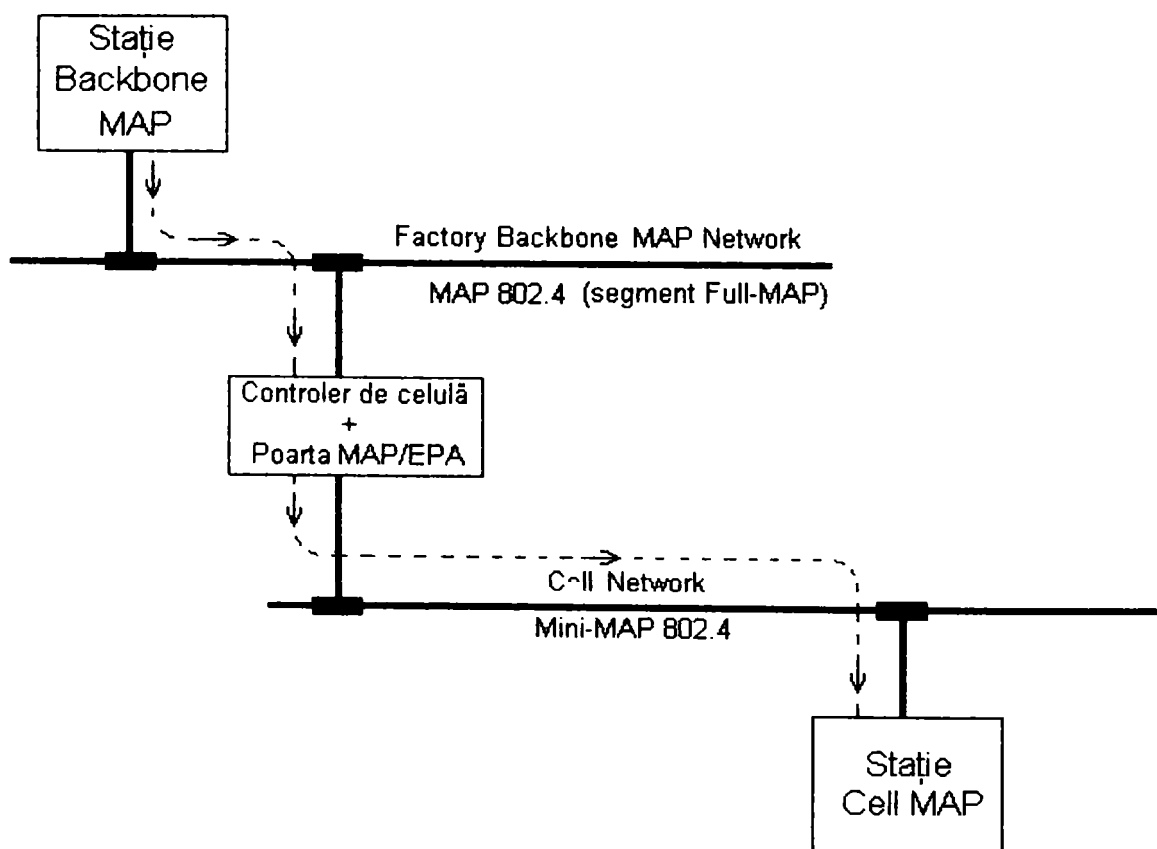


Fig.3.22. Transmiterea comenzilor între o stație din rețeaua coloană (backbone) cu arhitectură Full-MAP și o stație din cadrul unei celule de tip Mini-MAP, [6]

3.3.5.3. Succesele inițiale ale MAP-ului

După elaborarea specificațiilor, mai multe sute de întreprinderi s-au alăturat de proiectul MAP. Unele în calitate de producători de echipamente de fabricație (CNC, ROC, PLC), tab.3.2.

Nr. crt.	Furnizor	Țara	PLC	CNC	ROC
1.	ABB Cincinnati Group	Sweden /USA			T3-Control
2.	AEG Modicon	USA	A 984		
3.	Allen-Bradley	USA	PLC-3 PLC-5/250		
4.	April	France	5000 & 7000		
5.	Bosch	Germany	CL 500		
6.	Cincinnati Milacron	USA		Acrumatic-950	
7.	Fanuc	Japan		Series 15 Series 16	System R-H
8.	GEC Industrial Controls	UK	GEM80		
9.	GE Fanuc Automation	USA /Japan	Series 6 Series 90-70	Series 15 Series 16	
10.	GM Fanuc Robotics	USA /Japan			KAREL
11.	Grossenbacher	Switzerland		xNC	
12.	Kuka				RC
12.	Mitsubishi Electric	Japan	MELSEC A	MELDAS	
13.	Nippon Steel	Japan	PLC		
13.	NUM			1060 NC	
14.	Omron	Japan	SYSMAC		
15.	SattControl	Sweden	PLC		
16.	Siemens	Germany	SIMATIC S5	SINUMERIK	SIROTEK, ACR 20
17.	Telemecanique	France	TSX47	NUM	
18.	Texas Instruments	USA	Series 500		
19.	Toshiba	Japan	PLC		

Tab.3.2. Lista principalelor furnizori de echipamente de comandă dotate cu interfețe MAP, [8]

O altă categorie de întreprinderi simpatizante cu MAP au fost interesate în calitate de utilizatori. Aceste întreprinderi au practicat o filosofie după care "dacă e bun pentru GM atunci este bun și pentru noi", [6]. În tabele 3.3-3.6. s-a prezentat lista principalelor implementări ale MAP-ului pe plan mondial, [30, 31, 111].

Nr. Crt	Întreprinderea	Localitatea	Domeniul de fabricație
1.	Toyota Machine Works	Takahama-City, Japan	Car Parts Manufacturing
2.	Omron	Mashima-City, Japan	Electronic Parts Assembly
3.	Komatsu	Oyama-City, Japan	Press Machine Manufact.
4.	Mitsubishi	Nagoya-City, Japan	PLC Manufacturing Plant
5.	Toyota – Tahara Plant	Atsumi-gun, Japan	Car Assembly Plant
6.	Toyota – Motomachi - Plant	Toyota-City, Japan	Car Assembly Plant
7.	Toyota – Kamigoh – Plant	Toyota-City, Japan	Engine Plant

8.	Toyota – Takaoka – Plant	Toyota-City, Japan	Car Assembly Plant
9.	Toyota – Tsutsumi – Plant	Toyota-City, Japan	Car Assembly Plant
10.	Shinko	Hiroshima-ken, Japan	Machine Manufacturing
11.	Honda Giken	Tochigi-ken, Japan	Car Parts manufacturing
12.	KEPCO	Seoul, Korea	Electrical Power Plant

Tab.3.3. Principalele implementări MAP din Asia

Nr. Crt	Intreprinderea	Localitatea	Domeniul de fabricație
1.	GM Midsize Car Division	Oshawa, Ontario	Car Assembly Plant
2.	GM Saturn Corporation	Spring Hills, Tennessee	Car assembly Plant
3.	GM Truck Platforms	Indianapolis, Indiana	Metal Fabrication Plant
4.	GM Truck Platforms	Janesville, Wisconsin	Truck Assembly Plant
5.	GM Truck Platforms	Shreveport, Louisiana	Truck Assembly Plant
6.	GM Truck Platforms	Moraine, Ohio	Truck Assembly Plant
7.	GM Truck Platforms	Linden, New Jersey	Truck Assembly Plant
8.	GM Cadillac/ Lux. Car. Div.	Flint, Michigan	Car Assembly Plant
9.	GM Cadillac/ Lux. Car. Div.	Grand Blanc, Michigan	Metal Fabrication Plant
10.	GM Service Parts	Pittsburgh, Pennsylvania	Metal Fabrication Plant
11.	GM Truck Platforms	Pontiac, Michigan	Truck Assembly Plant
12.	GM Truck Platforms	Flint, Michigan	Metal Fabrication Plant
13.	GM Truck Platforms	Wentzville, Missouri	Truck Assembly Plant
14.	GM Lansing Automotive Div.	Lordstown, Ohio	Car Assembly Plant
15.	GM CPC-St. Catharines	St. Catharines, Ontario	Engine Assembly Plant
16.	GM Ypsilanti Powertrain	Ypsilanti, Michigan	Transmission Plant
17.	GM Saginaw Powertrain	Saginaw, Michigan	Foundry
18.	GM Delco Moraine	Moraine, Ohio	Brake Assembly
19.	GM Romulus Powertrain	Romulus, Michigan	Transmission Plant
20.	GM Bedford Powertrain	Bedford, Indiana	Foundry
21.	GM Powertrain	Windsor, Ontario	Transmission Plant
22.	ALCOA	Tennessee	Aluminium Plant
23.	Boeing Aerospace	Seattle, Washington	DNC System
24.	DuPont	Orange, Texas	Tank Car Loading System
25.	Grumman Aerospace	Long Island, New York	Parts Manufacturing Plant
26.	IBM	East Fishkill, New York	Ceramic Sintering Plant
27.	Martin Marietta	Oak Ridge, Tennessee	Command & Control
28.	Michelin Tires	Detroit, Michigan	Tire Manufacturing Plant
29.	NASA	New Mexico	Ground Tracking System
30.	Nissan Motors	Tennessee	Car Assembly Plant
31.	Ontario Water System	Ontario	Water Control System
32.	Union camp	Princeton, New Jersey	Paper Mill Project
33.	XEROX	Webster, New York	Copier Manufacturing

Tab.3.4. Principalele implementări MAP din America de Nord

Nr. Crt	Intreprinderea	Localitatea	Domeniul de fabricație
1	BHP Steel SPPD	Port Kambia, Australia	Tin Plating Control
2			Coke Oven Control
3			Cold Mill Control
4	BHP Steel RBBB	Newcastle, Australia	Power Sys. Distribution
5	Federal Airport Co.	Sydney, Australia	Airport & Building Mgmt.

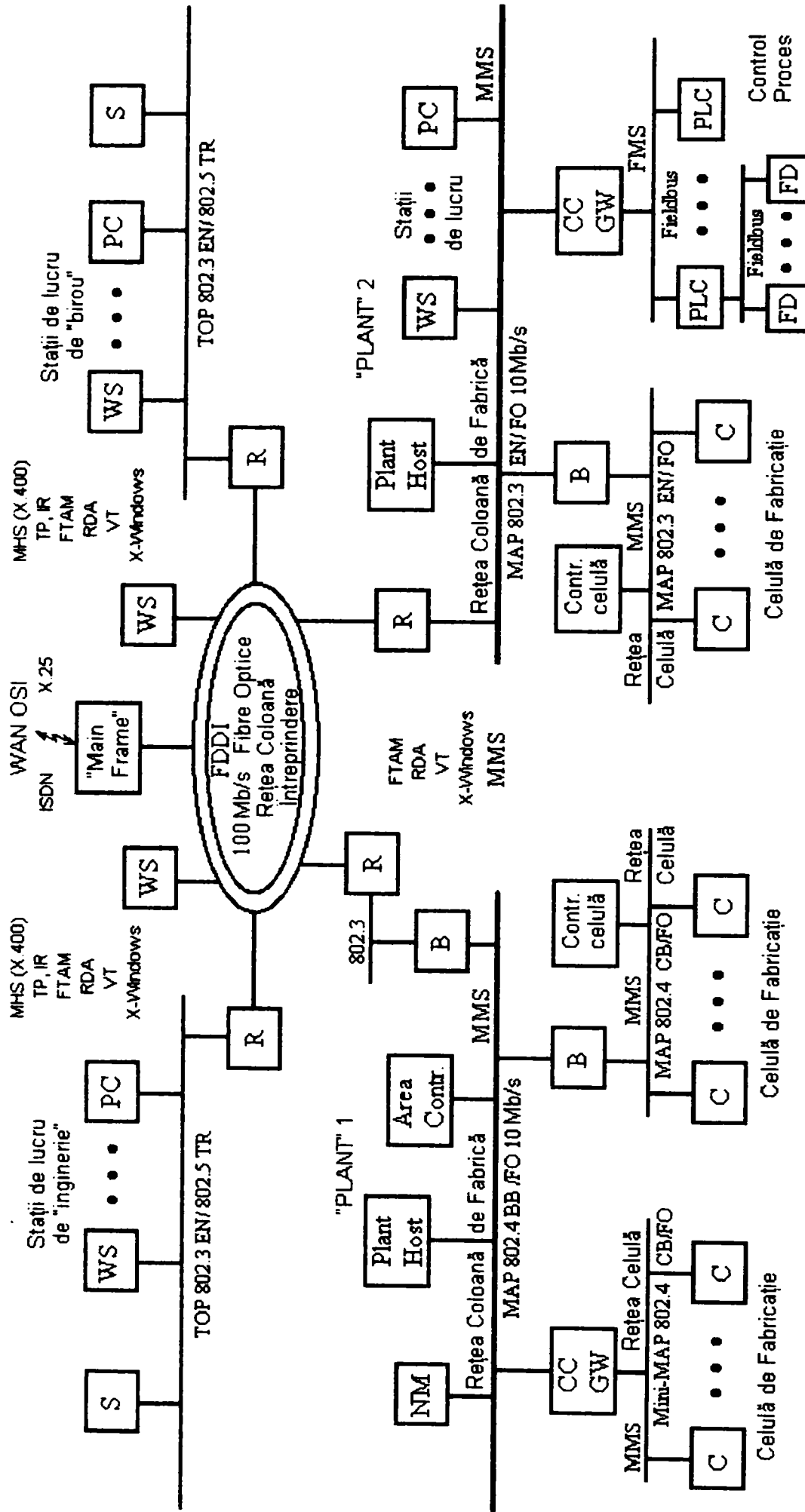
Tab.3.5. Principalele implementări MAP din Australia

Nr. Crt	Întreprinderea	Localitatea	Domeniul de fabricație
1	Renault	Flins, France	Car Assembly Plant
2	Renault	Cleons, France	Engine Plant
3	GM Vauxhall	Ellesmere Port, UK	Car Paint Unit
4	Volkswagen	Emden, Germany	Car Paint Unit
5	Volkswagen	Wolfsburg, Germany	Car Assembly Plant
6	Volkswagen	Hannover, Germany	Car Assembly Plant
7	Volkswagen	Mosel, Germany	Car Assembly Plant
8	Isuzu Bedford	Luton, UK	Light Truck Paint Unit
9	Henninger	Frankfurt, Germany	Brewery
10	Tuborg	Copenhagen, Denmark	Brewery
11	Kemira	Fredericia, Denmark	Fertilizer Plant
12	Mercedes-Benz	Sindelfingen, Germany	Car Parts Manufacturing
13	Renault	Le Mans, France	Engine Plant
14	Renault	Sandouville, France	Car Assembly Plant
15	Renault	Cleons, France	Engine Plant
16	BBA	Widnes, UK	Chemical Plant
17	KLV	Copenhagen, Denmark	Airport Runway Control
18	Pilz	Albrechts, Germany	CD Manufacturing Plant
19	Mercedes-Benz	Unterturkheim, Germany	Car Parts Manufacturing
20	Mercedes-Benz	Bremen, Germany	Car Assembly Plant
21	Mercedes-Benz	Zuffenhausen, Germany	Car Parts Manufacturing
22	EDF	Dunkerque, France	Power Plant
23	Elf	Donges, France	Oil Industry
24	Sidmar	Ghent, Belgium	Steel Plant
25	Rautaruuki Oy	Hameenlinna, Finland	Steel Plant
26	Statens Veevesen	Oslo, Norway	Highway Traffic Control
27	GM Opel	Eisenach, Germany	Car Assembly Plant
28	General Motors Spain	Saragoza, Spain	Car Assembly Plant
29	GM Vauxhall	Ellesmere Port, UK	Engine Plant
30	General Motors Hungary	Szentgothart, Hungary	Engine Plant
31	GM Opel	Russelsheim, Germany	Car Assembly Plant
32	GM Opel	Bochum, Germany	Car Assembly Plant
33	British Aerospace	England	Manufacturing Cell
34	BMW	Regensburg, Germany	Car Assembly Plant
35	Airtalia	Italy	Cable Assembly
36	ISW TU Stuttgart	Stuttgart, Germany	Manufacturing Cell
37	Magneti Marelli	San Salvo, Italy	Alternator Manufacturing
38	Aerospatiale	Chatillon, France	Missile Parts Manuf.
39	Renault	Paris, France	Network Management

Tab.3.6. Principalele implementări MAP din Europa

Acest avânt a dus mai departe proiectul față de primele imaginații. Firma Boeing, care era la fel de însemnat, a lansat proiectul TOP (Technical Office Protocol – *Protocol pentru Birouri Tehnice*) care este strâns legat de MAP și la care la fel s-a aderat un număr mare de firme producătoare și utilizatori, [94]. În Hipersisteme CIM, protocolul TOP a fost destinat interconectării deschise a calculatoarelor de la nivelele 1, 2 și 3. Specificațiile MAP 3.0 din 1993 chiar propune o structură informațională “MAP/TOP” pentru întreprinderile din domeniul Hipersistemelor CIM, [31].

În fig.3.23. s-a reprezentat această structura MAP/TOP pentru cazul unei întreprinderi cu mai multe platforme/ hale industriale (plant1 și plant2). Fiecare platformă are propria structură de comandă MAP, formată dintr-o rețea coloană de fabrică (Factory Backbone Network) respectiv din rețele de celulă (Cell Network).



BB – Bandă largă; CB – Cu "purțătoare"; FO – Fibre optice; EN – Bandă de bază; TR – Token Ring; B – Bridge; R – Router;
 GW – Gateway; WS – Stații de lucru; S – Server; CC – Controller de Celulă; C – PLC, CNC sau Controller Robot; FD – Dispozitive de "câmp"

Fig.3.23. Structura informațională MAP/TOP a unei întreprinderi conform specificațiilor MAP3.0 din 1993, [31]

Sunt prezentate structurile MAP respectiv modurile lor tipice de interconectare informațională. La fel este prezentat și modul de conectare la ele a unor structuri specifice domeniului proceselor industriale continue (Process Control).

Cele două structuri de comandă MAP, din cadrul celor două platforme/hale industriale, sunt conectate la structurile TOP din cadrul întreprinderii prin intermediul **Rețelei Coloană de Întreprindere** (Enterprise Backbone Network). Aceasta este o rețea cu fibre optice de tip FDDI (Fiber Distributed Data Interface) cu o mare capacitate de transmisie a informațiilor. Conectarea structurilor MAP și TOP la această rețea coloană de întreprindere se realizează prin intermediul "router"-elor.

Tot această rețea conectează la **sistemul informațional al întreprinderii** calculatorul central de întreprindere, numit: **Main Frame**. Acesta este un calculator de mare capacitate și are rolul coordonării tuturor activităților de producție din cadrul întreprinderii. La rândul său poate fi conectat la rețele WAN (Wide Area Networks), prin intermediul unor protocoale specifice comunicației la "distanță" (ISDN, X25), realizând astfel legături informaționale cu alte întreprinderi (aflate în calitate de furnizori și/sau de beneficiari) respectiv cu diferite instituții (bănci etc.).

3.3.6. Standardizarea comenzilor utilizate în cadrul structurilor de fabricație automatizată. Apariția MMS-ului

Odată cu standardizarea transmiterii comenzilor trebuia realizată și standardizarea mesajelor de fabricație aferente acestora.

3.3.6.1. MMS - primul standard deschis de comandă de la distanță

Standardizarea schimburilor de mesaje de fabricație în cadrul structurilor de fabricație automatizată s-a realizat sub forma unei specificații cunoscută sub numele de **Specificația Mesajelor de Fabricație** (Manufacturing Message Specification – MMS), [58, 59]. Aceasta conține referiri la sintaxa precum și la semantica mesajelor de fabricație respectiv fixează serviciile standarde ale echipamentelor de comandă, aferente acestor mesaje.

MMS-ul a fost dezvoltat inițial de asociația EIA (Electronic Industries Association) sub denumirea de **Formatul Standard al Mesajelor de Fabricație** (Manufacturing Message Format Standard – MMFS). MMFS a fost de fapt o anexa la specificațiile MAP versiunea 2, publicat în 1984. Acesta însă a permis prea multe "varianțe" de implementare. Astfel au rezultat un număr mare și respectiv incompatibile "dialecte" ale MMFS.

În consecință a apărut necesitatea elaborării unui "standard al mesajelor de fabricație" mult mai riguros. Acesta a fost elaborat de către **Organizația Internațională pentru Standardizare** (International Organization for Standardization), sub denumirea de **"ISO/IEC 9506, Industrial Automation Systems - Manufacturing Message Specification (MMS)"**, respectiv utilizează un sistem de indici respectiv de codificare internațională (ASN.1), [27, 76].

Ca parte a specificațiilor MAP, MMS-ul este implementat la nivelul 7, fiind interfața directă cu "aplicația industrială", fig.3.24., [1].

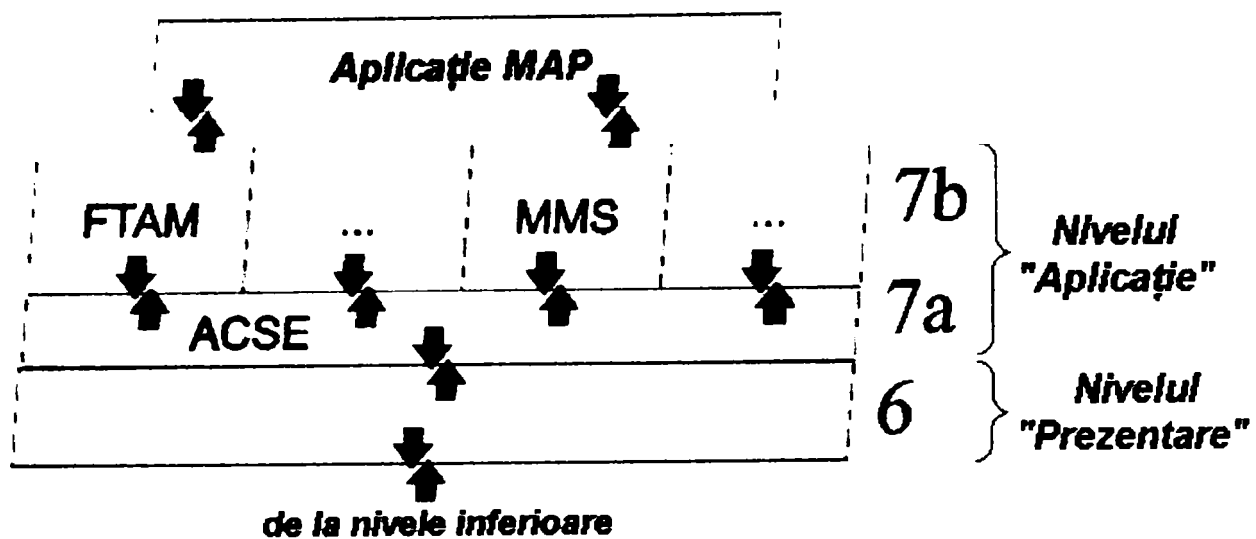


Fig.3.24. Rolul MMS-ului în cadrul MAP-ului: interfața cu aplicația industrială

Trebuie menționat faptul că MMS-ul nu este o simplă colecție de mesaje de fabricație (respectiv servicii) standardizate. MMS-ul este mult mai mult. MMS-ul este un adevărat *limbaj de fabricație automatizată* (manufacturing automation language), care în cadrul Hipersistemului CIM este utilizat pentru realizarea programelor de comandă și respectiv de monitorizare a structurilor de fabricație automatizate. În prezent există deja o serie de furnizori care oferă medii software pentru dezvoltarea aplicațiilor MMS, tab.3.7.

Nr. crt.	Furnizor	Denumire program
1.	Cycle	LiveData
2.	Digital	BaseStar Open
3.	GE Fanuc Automation	Cimplicity
4.	HP	RTAP/PLUS
5.	Mercedes Benz	AIMS
6.	Prism Technologies Limited	OpenBase
7.	Silicomp, Bull	Appli-Bus
8.	Siemens	TF/DDE Manager
9.	Siemens	SICALIS PMC
10.	SISCO	AX-S4 MMS
11.	SISCO	MMS LINK
12.	SNI	SINEC MMS-I Variable Generator
13.	VM-data A/S	EasyMAP®

Tab.3.7. Lista principalelor medii software destinate dezvoltării aplicațiilor MMS, [19]

MMS-ul asigură "utilizatorilor" posibilitatea concentrării asupra aplicației propriu-zise (adică asupra procesului de fabricație), și nu asupra transmiterii mesajelor de fabricație. Această transmitere a mesajelor se realizează automat, respectiv în mod transparent, de către celelalte "nivele" OSI.

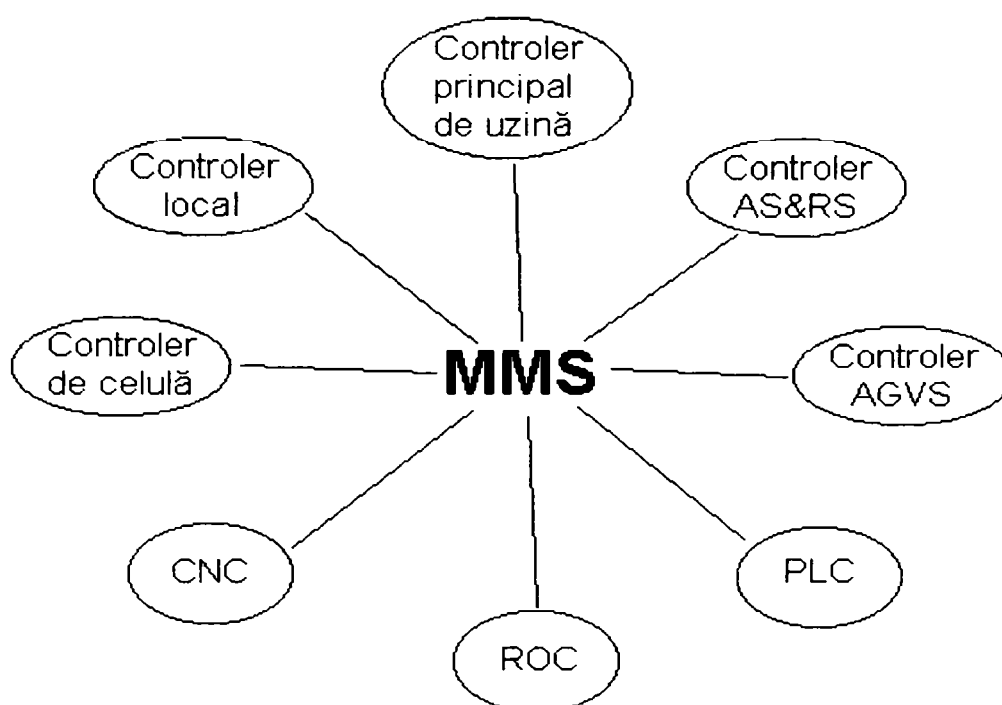


Fig.3.25. MMS-ul este un limbaj comun pentru toate echipamentele de comandă

Pe de altă parte, așa cum rezultă din standardul ISO/IEC 9506, partea 1. și 2., MMS-ul este un **limbaj independent de echipament**. Adică, un limbaj ce poate fi "înțeles", cel puțin formal, de fiecare dintre echipamentele de comandă din cadrul Hipersistemului CIM. Astfel MMS-ul este limbajul comun, atât pentru controlerile de celulă, controlerul AS&RS, controlerul AGVS, controlerile locale sau controlerul principal de uzină cât și pentru echipamentele de tip CNC, ROC sau PLC, (fig.3.25.).

Astăzi MMS-ul este considerat un **standard deschis de comandă de la distanță** independent de mediul de transmitere a mesajelor. Aceasta face posibilă implementarea MMS-ului pe diferite platforme și sisteme de operare independent de structurile MAP, [34, 104].

3.3.6.2. Structura funcțională a MMS-ului. Echipamentul Virtual de Fabricație

D.p.d.v. funcțional MMS-ul este o **interfață între echipamentele interconectate**. În cazul echipamentelor de comandă (CNC, PLC, ROC etc.), interfața MMS permite accesul la funcțiile și resursele acestora. Utilizând o arhitectură client-server, acest acces se realizează prin intermediul **serviciilor MMS** care pot fi accesate din exterior prin intermediul unor "funcții" standard, numite **cereri de servicii**.

Practic sunt definite, două tipuri de interfețe MMS. O interfață specifică echipamentelor de fabricație, care oferă servicii standardizate, pentru accesul de la distanță la funcțiile și resursele acestora și o interfață, specifică controlerelor amintite, care în scopul efectuării accesului amintit (necesar pentru comandă și monitorizare de la distanță) realizează emiterea cererilor de servicii. Primul tip de interfață, adică cea care oferă servicii către celălalt tip, este pe post de **furnizor de servicii** și în

consecință este numită **interfață MMS Server**. Celălalt tip este pe post de **utilizator de servicii** și în consecință este numită **interfață MMS Client**, fig.3.26.

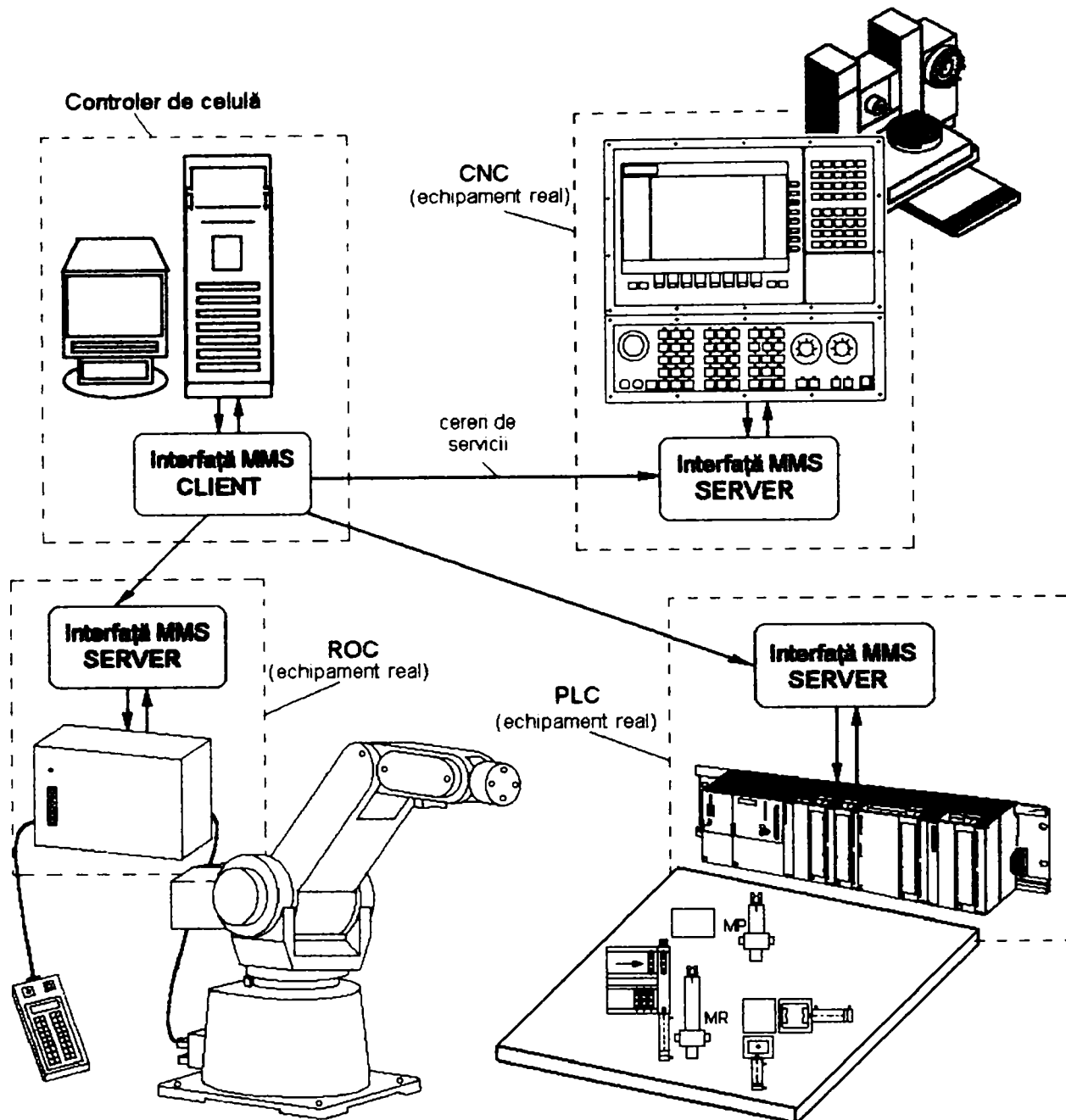


Fig.3.26. Rolul celor două tipuri de interfețe MMS în cadrul unei celule de fabricație

În schimburile de mesaje de fabricație, echipamentul de comandă (CNC, PLC, ROC etc.) care este dotat cu interfață MMS Server este numit: SERVER, iar calculatorul pe post de controler de celulă, stație operator etc., dotat cu interfață MMS Client, are denumirea de CLIENT.

Standardul ISO9506 se referă doar la interfața MMS Server, [58, 59]. Adică sunt standardizate doar serviciile care trebuie să fie îndeplinite de către echipamentele de comandă din cadrul sistemelor industriale automate respectiv stabilește și funcțiile standard de acces la aceste servicii.

Conform standardului, accesul la funcțiile și resursele unui echipament real de fabricație se realizează indirect, prin intermediul unui model abstract al echipamen-

tului real, numit: **Echipament Virtual de Fabricație** (Virtual Manufacturing Device - VMD), fig.3.27.

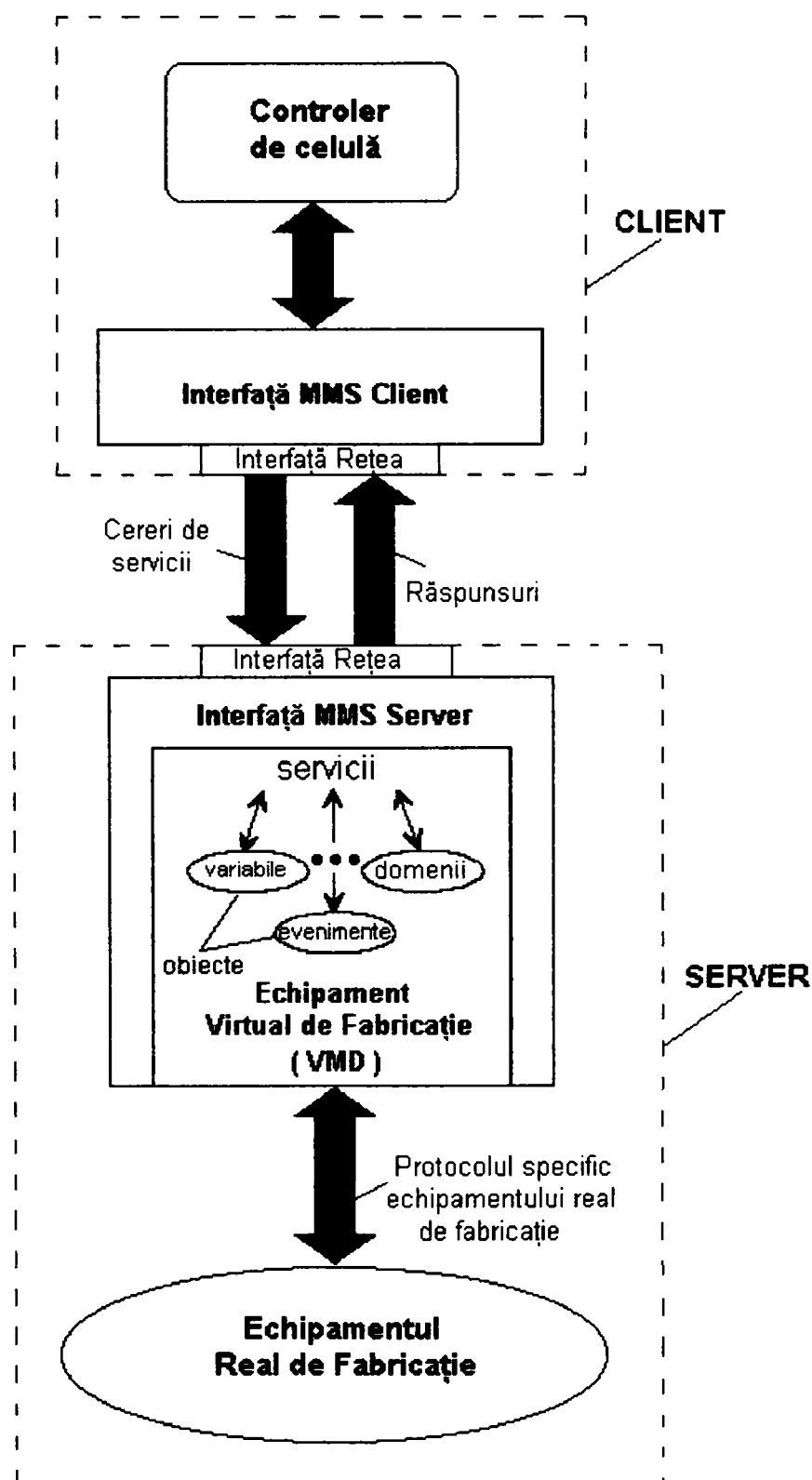


Fig.3.27. Rolul Echipamentului Virtual de Fabricație, [102, 103]

Acesta conține o descriere abstractă și standardizată a funcțiilor și resurselor echipamentului real, indiferent de tipul și de proveniența acestuia. Ca urmare, din "exterior", fiecare echipament real este "observat" sub forma unui Echipament Virtual de Fabricație, [45]. Toate cererile de servicii, emise de exemplu de controlerul de celulă (prin intermediul interfeței MMS Client), sunt adresate Echipamentului Virtual

de Fabricație, care apoi are sarcina ducerii la buna îndeplinire a serviciilor solicitate prin legătura directă și exclusivă cu echipamentul real.

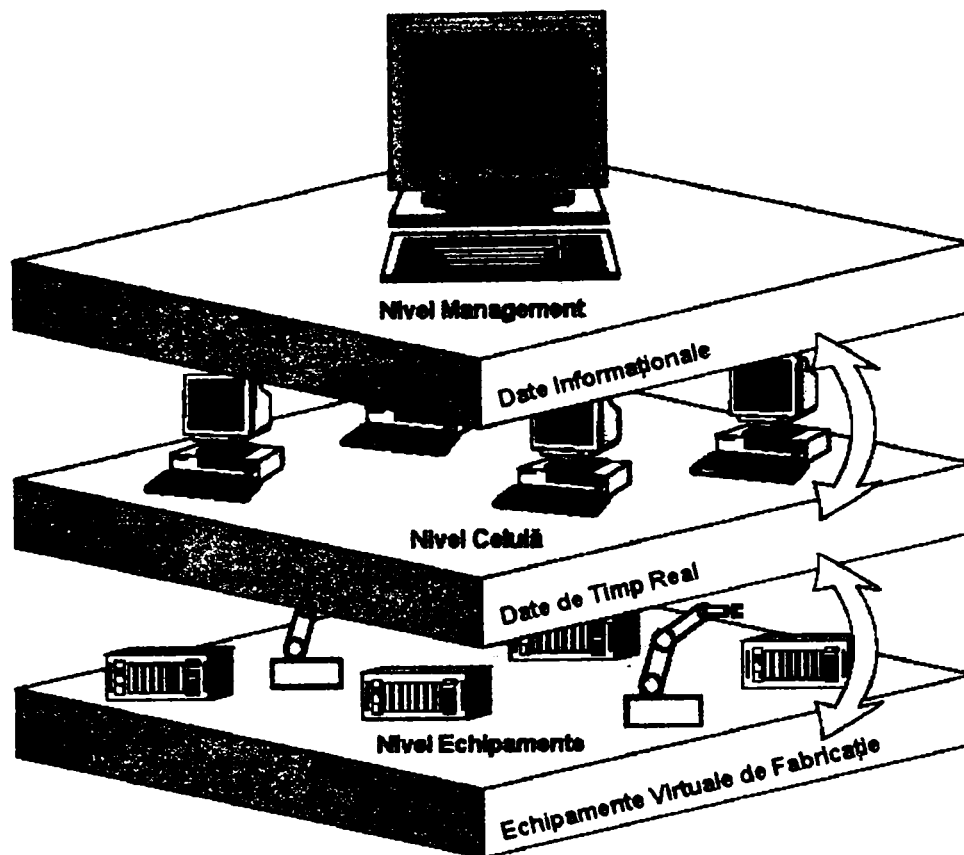


Fig.3.28. Fluxul informațional între 3 nivele ale Hipersistemului CIM

Pe baza celor enunțate, prin intermediul cererilor de servicii sunt accesate practic echipamentele virtuale de fabricație și nu cele reale. Fig.3.28. Încearcă să pună în evidență acest aspect prin reprezentarea fluxurilor informaționale existente între 3 nivele de comandă din cadrul Hipersistemului CIM, [107]. Controlerile de celulă, aflate la nivelul celulelor (cell level) accesează echipamentele virtuale de fabricație de pe nivelul echipamentelor (device level).

Echipamentul Virtual de Fabricație utilizează pentru descrierea abstractă a funcțiilor și resurselor echipamentului real, modele abstracte numite **obiecte MMS**, [87]. Aceste obiecte reprezintă câte o resursă fizică sau logică a echipamentului real.

Denumire obiect MMS	
Virtual Manufacturing Device	Echipament Virtual de Fabricație
Domain	Domeniu, resursă
Program Invocation	Invocație Program
Variable (+ List)	Variabilă (+ Listă)
Type	Tip
Event	Eveniment
Journal	Jurnal

Tab.3.8. Principalele obiecte și servicii MMS

Tabelul 3.8. conține principalele obiecte MMS care sunt independente de tipul echipamentului de comandă, [88]. Accesul la obiecte se realizează prin **serviciile MMS**. Aceste servicii oferă astfel acces la resursele echipamentului real, [106].

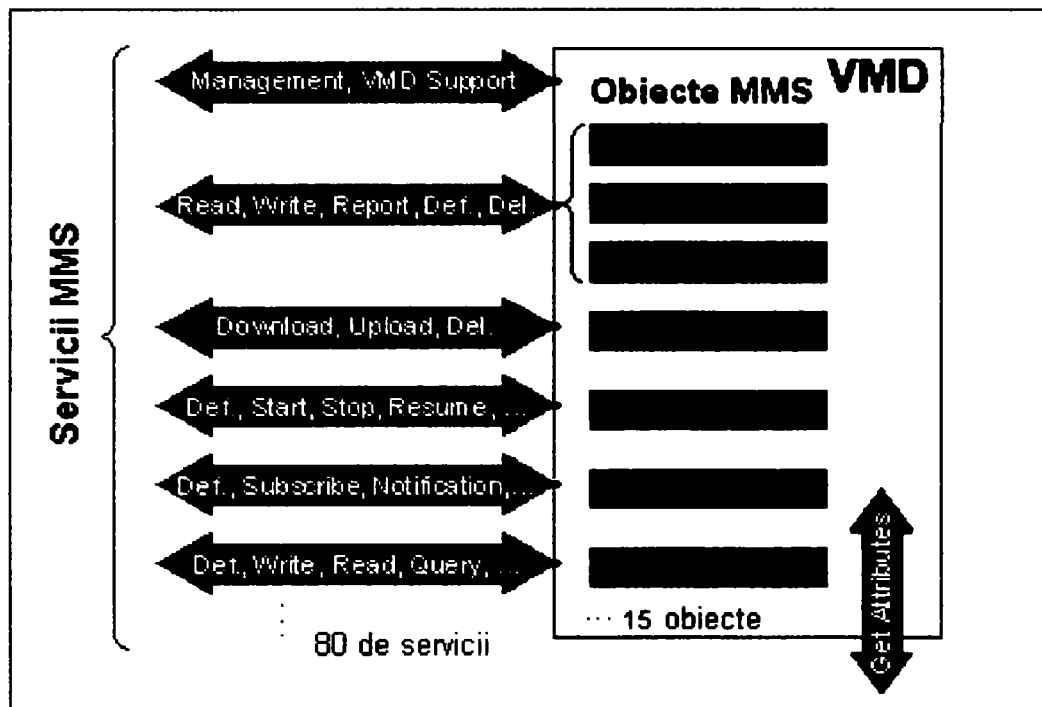


Fig.3.29. Accesul la obiecte prin servicii MMS

Prin intermediul **invocațiilor de program** (Program Invocation) se poate prelua controlul asupra programelor de comandă din cadrul echipamentului real. Aceste programe, prin serviciile aferente, pot fi lansate în execuție (Start), pot fi oprite (Stop), pot fi relansate (Resume) etc. De asemenea, prin servicii MMS se poate obține acces la o serie de **variabile, date și liste** legate de aplicație sau de procesul condus, [106]. Orice **eveniment** produs în timpul funcționării poate fi notificat (Notification) prin intermediul unui serviciu MMS. Pentru înregistrarea cronologică a informațiilor cum sunt: evenimente, valorile variabilelor sau texte referitoare la observații (commentarii) operator se utilizează obiecte de tip **jurnal** (Journal). Înregistrările din cadrul jurnalului pot fi citite cu ajutorul serviciului de **citire** (Read).

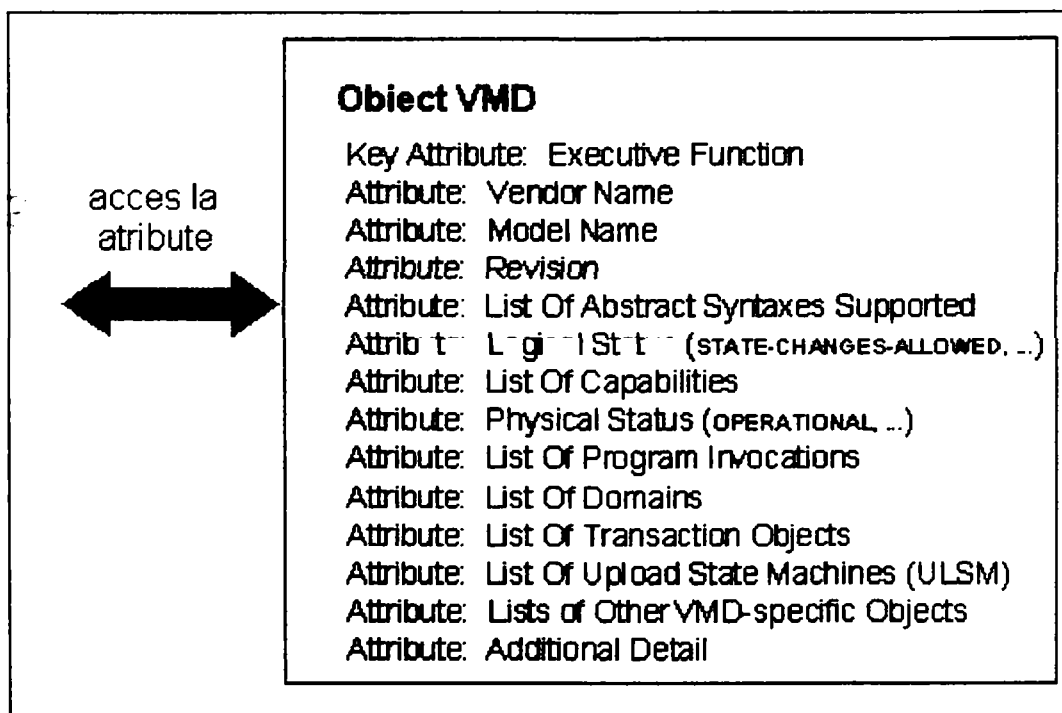


Fig.3.30. Accesul la attributele obiectelor

Cu ajutorul serviciilor MMS se poate realiza acces și la conținutul atributelor unui obiect (Get Attributes), fig.3.29. Astfel, de exemplu în cazul obiectului VMD, se pot obține informații despre **funcția executivă** (Executive Function) a echipamentului adică despre funcția principală pentru care a fost creat echipamentul real (de exemplu de comandă robot), [103]. De asemenea se pot afla informații despre **denumirea dată de furnizor** (Vendor Name), **numele modelului** (Model Name) și **număr serie** (Revision). Pentru monitorizarea de la distanță a stărilor de funcționare este nevoie de acces la attribute prin care sunt descrise **stările logice** (Logical Status) și **stările fizice** (Physical Status). Între altele se poate obține și câte o listă despre numele celorlalte obiecte (invocații program, domenii etc.) din cadrul VMD-ului în vederea accesării lor cu serviciile aferente.

În cazul unor echipamente concrete de comandă, se poate avea acces la o serie de **attribute adiționale** (Additional Detail) prin care sunt descrise resursele speciale ale echipamentului real.

3.3.6.3. Încercări privind elaborarea unor structuri MMS specifice pentru fiecare tip de echipament de fabricație

Pentru ca mesajele de fabricație să fie "înțelese" în mod echivoc, de către fiecare echipament de comandă, nu numai formal dar și în conținutul lor, a fost începută elaborarea unor specificații adiționale pentru fiecare tip de echipament de comandă din cadrul structurilor de fabricație automatizate. Prin intermediul acestor specificații, s-a încercat standardizarea unui număr minim de mesaje de fabricație (respectiv servicii), corespunzător funcțiilor specifice ale fiecărui tip de echipament de comandă în parte, [46].

Până în prezent au fost elaborate specificații adiționale MMS, pentru următoarele echipamentele de comandă:

1. Echipamente de comandă robot (ROC – Robot Controller);
2. Echipamente de comandă numerică (CNC- Computer Numerical Control);
3. Automate programabile (PLC –Programable Logic Controller);
4. Echipamente de conducere a proceselor industriale (CPC – Computerized Process Control).

Aceste specificații au fost elaborate sub forma unor standarde adiționale la standardul ISO/IEC 9506, deci nu sunt menite să înlocuiască ci să completeze pe acesta. Realizând aceste standarde adiționale, inevitabil au apărut în plus o serie de mesaje de fabricație respectiv servicii specifice care nu erau conținute în standardul de bază sau erau în contradicție cu acesta. Au apărut astfel diferențe semnificative în limbajul de comandă utilizat la diferite echipamente de comandă (de exemplu între ROC și CNC).

În prezent se încearcă **integrarea tuturor serviciilor adiționale în standardul ISO9506 de bază**. În consecință majoritatea standardelor adiționale au fost retrase (de exemplu și standardul ISO9506-3 referitor la roboți industriali).

Ultima versiune a standardului ISO9506 a fost publicat în data 01.01.2002.

3.3.7. Necesitatea elaborării unor variante cu cost redus ale structurilor deschise de comandă utilizate în Hipersisteme CIM

Așa cum s-a arătat în cadrul paragrafelor anterioare, prin intermediul structurilor deschise de comandă, rezultate din implementarea specificațiilor MAP și respectiv MMS, **pot fi rezolvate toate problemele tehnice legate de automatizarea fabricației** în cadrul Hipersistemelor CIM.

Astfel, acolo unde este necesară garantarea unui control în timp real a fabricației poate fi utilizată arhitectura Mini-MAP (organizată pe 3 nivele OSI). Iar acolo unde este necesară asigurarea caracterului deschis al structurii de comandă (în vederea asigurării compatibilității cu alte structuri OSI), se poate utiliza arhitectura Full-MAP (organizată pe 7 nivele OSI).

Pentru structurile de fabricație cu trafic intens de date, MAP oferă posibilitatea transmiterii comenzilor într-un mediu cu bandă largă (Broadband). Pentru structurile de fabricație cu trafic redus de date oferă posibilitatea utilizării unor structuri de comandă mai ieftine, care au la bază o metodă de acces la mediu utilizat în cadrul așa-numitor rețele Ethernet.

Prin promovarea pe scară largă a structurilor deschise, rezultate din implementarea specificațiilor MAP, se putea rezolva definitiv și problema costurilor de integrare generată de incompatibilitatea sistemelor de comandă. Însă nu s-a întâmplat așa. Probabil, din motivul că nu s-a reușit realizarea și fabricarea în serie a unui "chip electronic" dedicat, care să conțină o implementare hardware a întregului protocol MAP. În aceste condiții, fiecare fabricant de echipamente, trebuia să facă un **efort suplimentar** pentru implementarea specificațiilor MAP în produsele lor proprii. Acest efort suplimentar bineînțeles s-a regăsit și în prețul de cost al acestor echipamente.

Astfel structurile de comandă care utilizau interfețe MAP, deși aveau la bază standarde deschise, **au devenit mai scumpe** decât cele care utilizau interfețe cu protocoale brevetate (proprietary). În aceste condiții, utilizarea lor era economică numai în acele întreprinderi mari, în special din industria automobilelor, în care frecvent apărea necesitatea integrării în fabricație a unor echipamente noi. În aceste întreprinderi, prețurile mai ridicate ale acestor structuri de comandă erau compensate de costurile minime de integrare a lor informațională. În celelalte întreprinderi însă, utilizarea lor nefiind economică, au fost promovate alte tipuri de structuri de comandă, mai mult sau mai puțin deschise, dar mai ieftine.

În aceste condiții, pentru menținerea avantajelor oferite de sisteme deschise de comandă (referitor la integrare informațională), a apărut necesitatea elaborării și introducerii în Hipersisteme CIM, a unor variante ale acestora **cu cost redus**, [44].

3.4. Introducerea în Hipersisteme CIM a unor structuri deschise de comandă cu cost redus

Pentru introducerea în Hipersisteme CIM a unor structuri deschise de comandă cu cost redus există două mari tendințe. Pe de o parte se încearcă introducerea unor protocoale ale "internetului", care în timp au devenit standarde "de facto" și ast-

fel utilizarea lor este foarte ieftină. Pe de altă parte se încercă introducerea în Hipersisteme CIM a unor structuri deschise de comandă, care inițial au fost destinate domeniului proceselor industriale (unde se utilizează cu succes și în prezent).

3.4.1. Introducerea unor structuri deschise ale “internetului” în Hipersisteme CIM

O posibilitate pentru obținerea în cadrul Hipersistemelor CIM a unor modalități de comandă de la distanță, în variantă cu cost redus, este utilizarea unor structuri deschise oferite de “suita de protocoale” TCP/IP, [43]. Această suită de protocoale, actualmente stă la baza realizării interconectărilor, din cadrul rețelei mondiale “internet”.

Odată cu răspândirea spectaculoasă și în același timp generală a internetului, suita de protocoale TCP/IP a fost unanim acceptată, atât de către utilizatorii cât și de producătorii de software, astfel încât în scurt timp ea a devenit standard “de facto”. Specificațiile fiind publice, un număr relativ mare de producători (de software pentru calculatoare) realizează deja implementarea acestora în produsele lor proprii. Chiar există deja programe software comerciale oferite în mod gratuit sau la un preț mic. În aceste condiții, implementarea acestor protocoale în echipamentele de fabricație din cadrul Hipersistemului CIM, conduce în mod sigur la obținerea unor structuri de comandă de la distanță cu cost redus. De asemenea, se rezolvă în același timp și problema costurilor de integrare a echipamentelor noi (bineînțeles, cu condiția ca și ele să utilizeze pentru schimburile informaționale suita de protocoale TCP/IP).

3.4.1.1. Implementarea MMS-ului sub TCP/IP. Specificația firmei Boeing

Implementarea MMS-ului sub TCP/IP a fost propusă inițial de **Grupul European de Utilizatori MAP** (European MAP Users Group – EMUG) în scopul obținerii unor variante mai ieftine pentru structurile MAP.

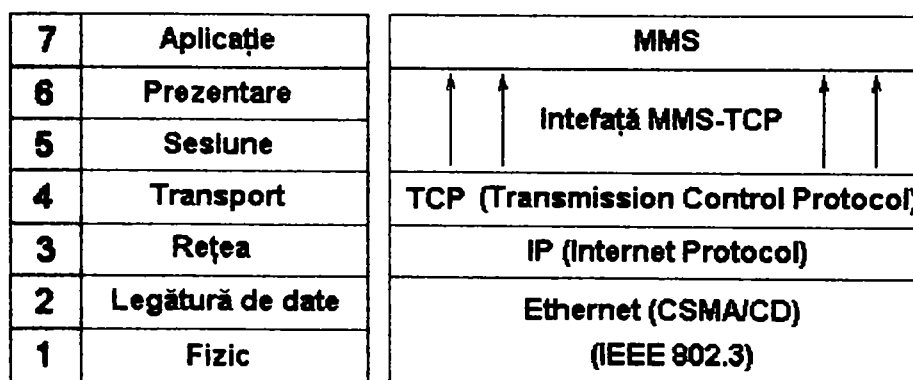


Fig.3.31. Arhitectura MMS-TCP/IP, [42]

Mentținând MMS-ul pe ultimul nivel (de aplicație) s-a propus de asemenea utilizarea pe primele două nivele, a standardului “Ethernet” (IEEE 802.3) pentru accesul la mediul fizic. Aceasta asigură o soluție mai ieftină decât cea oferită de standardul IEEE 802.4. aferent metodei de acces cu jeton (Token Bus).

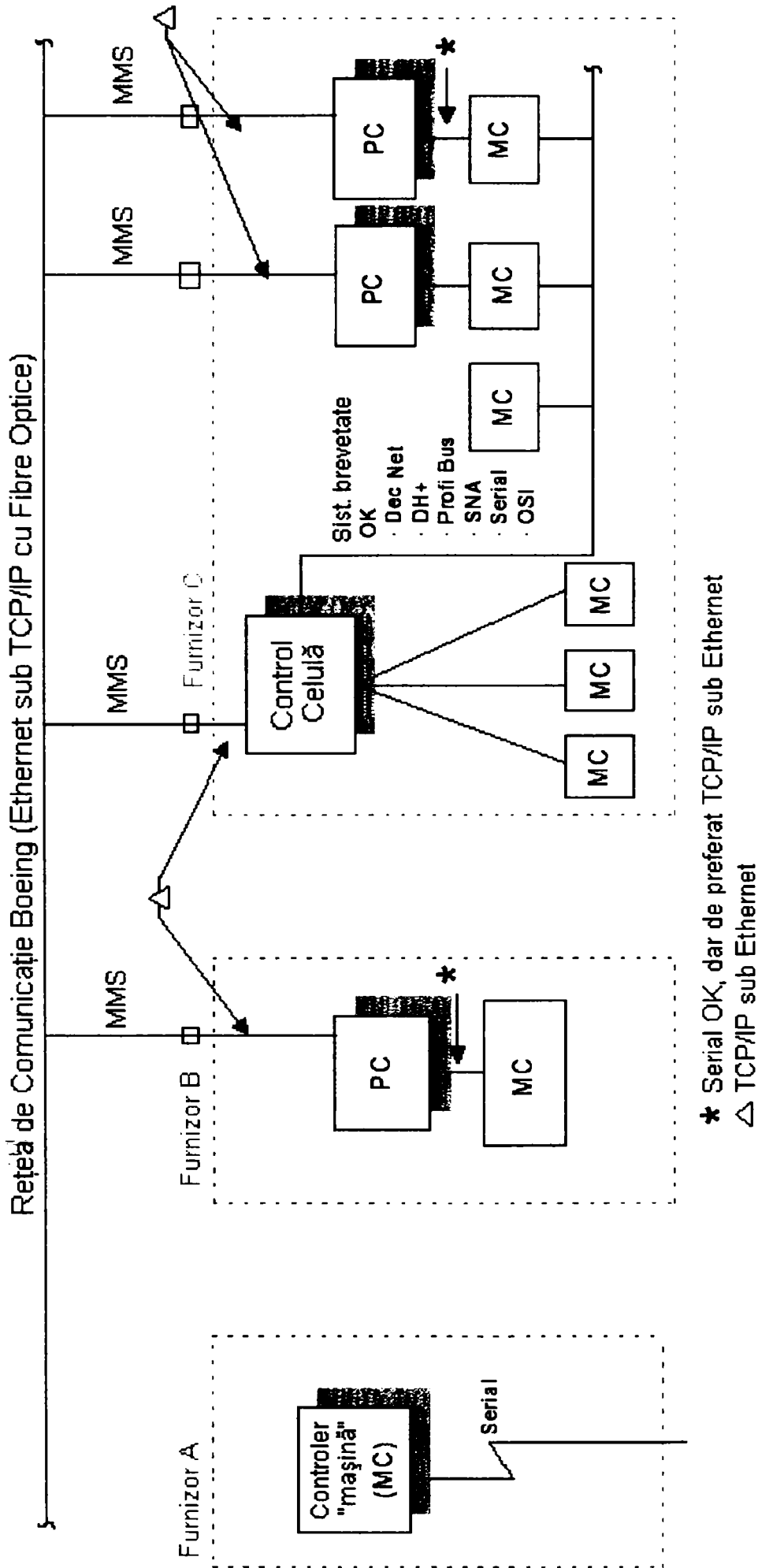


Fig. 3.32. Specificațiile firmei Boeing referitor la interconectările informaționale ale echipamentelor noi de fabricație, [17]

La nivelul 3 (rețea) este utilizat protocolul *IP (Internet Protocol)* care asigură dirijarea (routing) a pachetelor de date între rețelele LAN interconectate și de asemenea identificarea adresei destinatarului, [95].

Observație:

- Fiecare stație din rețelele interconectate, are o adresă unică (numită adresă IP) care este formată din 4 grupe de cifre între 0 și 255 (de exemplu: 192.168. 0. 34.), [95]. Această adresă conține codificat atât numele (numărul) rețelei din care face parte stația precum și un identificator unic al stației respective.

Cu controlul transportului pachetelor de date între stația sursă și stația destinație se ocupă protocolul *TCP (Transmission Control Protocol)* de pe nivelul 4. Spre deosebire de modelul OSI, nivelele 5 și 6 nu sunt utilizate. Însă, pentru legătura cu MMS-ul implementat pe nivelul 7 de aplicație este nevoie de o interfață MMS-TCP. Această arhitectură permite obținerea unor structuri de comandă de la distanță cu cost redus, însă nu garantează în toate cazurile efectuarea unui control în timp real.

Un exemplu tipic pentru promovarea unor structuri de comandă cu cost redus, prin utilizarea MMS-ului sub TCP/IP, este **cazul firmei Boeing**. Aceasta, în calitate de utilizator, în **anul 1998** a formulat o serie de specificații tehnice, referitor la modalitățile schimburilor informaționale ale echipamentelor de fabricație, care urmează să fie integrate în structurile de fabricație ale firmei. În consecință, va achiziționa numai acele echipamente care respectă specificațiile respective.

Conform fig.3.32. firma Boeing, a stabilit cerințe pentru 3 cazuri diferite. În cazul (A) în care un furnizor oferă echipamente mici (small) de fabricație, atunci ele trebuie să fie dotate cu interfețe seriale compatibile cu cele ale altor câțiva furnizori. În cazul (B) în care un furnizor oferă un echipament complet de fabricație dar independent (adică care nu este integrat într-o celulă de fabricație), atunci acesta trebuie să permită o conexiune cu sistemul de comandă general al firmei Boeing, printr-o legătură TCP/IP sub Ethernet, respectiv cu MMS. Dacă echipamentul de comandă este interfațat cu un calculator, atunci se admite o legătură serială între ele, însă este preferată o conexiune cu TCP/IP sub Ethernet.

În ultimul caz (C), când un furnizor oferă "la cheie" o celulă de fabricație cu mai multe echipamente de fabricație, atunci se cere ca legăturile externe ale celulei prin care se integrează în structurile de fabricație ale firmei Boeing, să fie realizate prin TCP/IP sub Ethernet, respectiv cu MMS. Legăturile interioare pot fi realizate chiar și cu protocoale brevetate, în condițiile unor prețuri rezonabile.

3.4.1.2. Proiectele "Wave I - III" ale firmei General Motors

În **aprilie 2000** firma General Motors, în colaborare cu firma Ford, a lansat prin proiectele "Wave" I, II și III., o strategie de includere treptată în rândul standardelor de comunicație, destinate roboților industriali, a unor protocoale ale suitei TCP/IP, [26, 110]. S-a considerat că aceste protocoale, implementate în cadrul controlerelor de robot, vor facilita o **integrare cu cost redus** a acestora în structurile de fabricație din cadrul întreprinderilor.

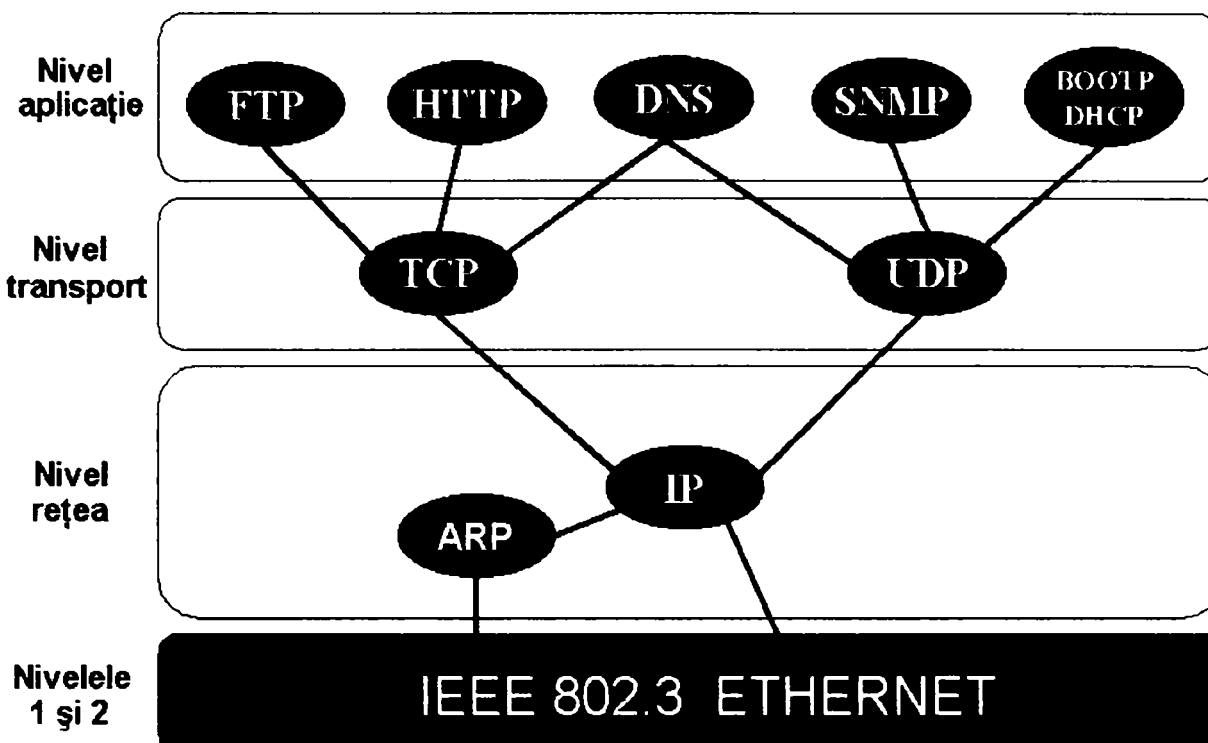


Fig.3.33. Principalele structuri TCP/IP promovate prin proiectele Wave I, II și III, [35]

În fig.3.33. s-au reprezentat principalele protocoale ale suitei TCP/IP care sunt prevăzute a fi implementate în sistemele de comandă ale roboților industriali. **Funcțiile generale** ale acestora sunt următoarele:

- **FTP** (File Transfer Protocol) va asigura o încărcare/descărcare standardă a programelor de aplicație în/din controlerul de robot.
- **HTTP** (HyperText Transport Protocol) va asigura un acces de la distanță la resursele robotului, în mod standard, asemănător navigării pe internet.
- **DNS** (Domain Name System) va oferi utilizarea în locul adreselor IP, a unor denumiri consacrate pentru stațiile interconectate (de exemplu *kuka117.gm* în locul *192.168.0.34*).
- **SNMP** (Simple Network Management Protocol) va fi utilizat pentru raportarea în mod standard a evenimentelor și a alarmelor aferente erorilor apărute în cadrul rețelei.
- **ARP** (Address Resolution Protocol) va preveni erorile cauzate de introducerea manuală a unor adrese incorecte.
- **DHCP** (Dynamic Host Configuration Protocol) va asigura alocarea automată a adreselor IP în rețea în locul alocării manuale, care este foarte costisitoare de timp în cazul în care numărul stațiilor este mare. (Platformele industriale ale firmei General Motors conțin în medie 300-600 de roboți industriali)
- **BOOTP** (Bootstrap Protocol) va oferi o reîncărcare automată a controlerului de robot, cu programe și date, după o eventuală "cădere a rețelei", [26]. Astfel se poate realiza o reinițializare rapidă și fără erori a structurilor de comandă. În opinia lui General Motors această funcție este foarte importantă în condițiile în care, la această firmă, o oprire accidentală a sistemului cauzează pierderi în medie de 10.000 \$ pe minut.

Pentru primele două nivele s-a recomandat implementarea standardului IEEE 802.3 în variantă de "*Ethernet 10/100*", cu conector RJ-45, [35].

Se considerat că dacă un controler de robot va avea implementate aceste structuri și va fi conectat la rețeaua informațională a întreprinderii, atunci accesarea de la distanță a resurselor fizice și logice va fi posibilă prin intermediul unor sisteme de programe comerciale devenite între timp standarde "de facto". De exemplu sistemul de operare *Windows 95/NT/2000* sau *Internet Explorer*-ul din cadrul acestuia.

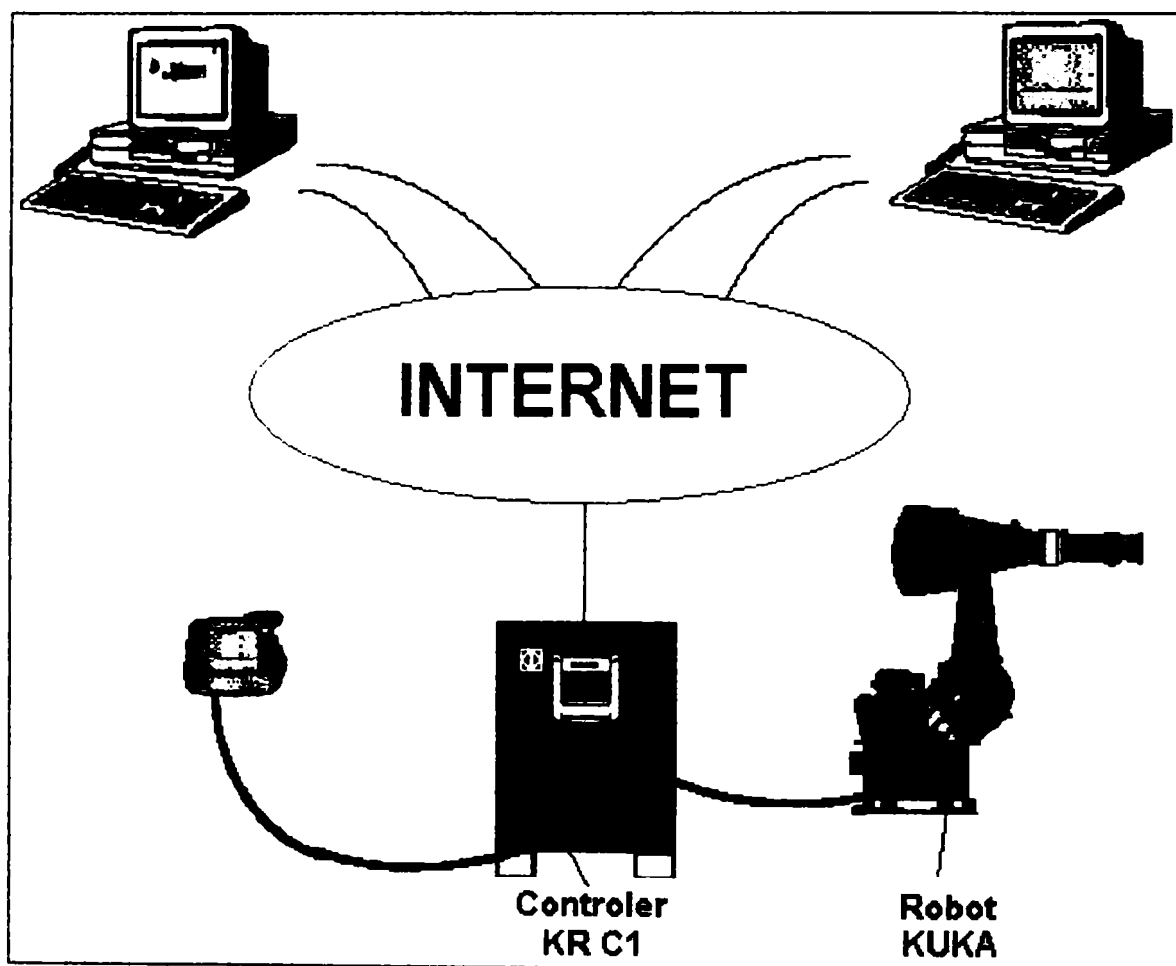


Fig.3.34. Diagnosticarea prin internet a defecțiunilor robotului KUKA, [74]

O eventuală **conectare la rețeaua internet**, a structurilor de comandă din cadrul Hipersistemului CIM, va face posibilă o intervenție mai rapidă a firmelor de servicii, în cazul unor funcționări eronate ale acestora. Astfel va fi posibilă de exemplu, diagnosticarea de la distanță, respectiv prin internet, a unor defecțiuni apărute la echipamentele de comandă (CNC, ROC, PLC) și eventuala remediere a lor prin reîncărcarea de la distanță a software-ului de bază, fig.3.34.

Trebuie arătat faptul că încercarea introducerii structurilor TCP/IP în Hipersisteme CIM, nu are ca scop primordial realizarea unui acces prin internet la controlul resurselor echipamentelor de fabricație.

Se urmărește însă, implementarea la nivelul halei industriale, a unui **mediu standard, deschis și ieftin de transmitere a informațiilor**. Aceasta va sta la baza realizării unor structuri de comandă cu cost redus (prin promovarea utilizării programelor comerciale disponibile).

3.4.1.3. Specificația FL-net a grupului JOP din Japonia

O altă încercare pentru dezvoltarea unor sisteme de comandă deschisă, pentru Hipersisteme CIM, este **specificația FL-net** a Grupului de Promovare a Sistemelor Deschise din Japonia (Japan FA Open Systems Promotion Group – JOP). Această specificație, elaborată în anul 1999, vizează realizarea unei interconectări a echipamentelor CNC, ROC, PLC și calculatoare, prin utilizarea unui set mai restrâns de protocoale ale internetului, în varianta UDP/IP (în loc de TCP/IP), fig.3.35.

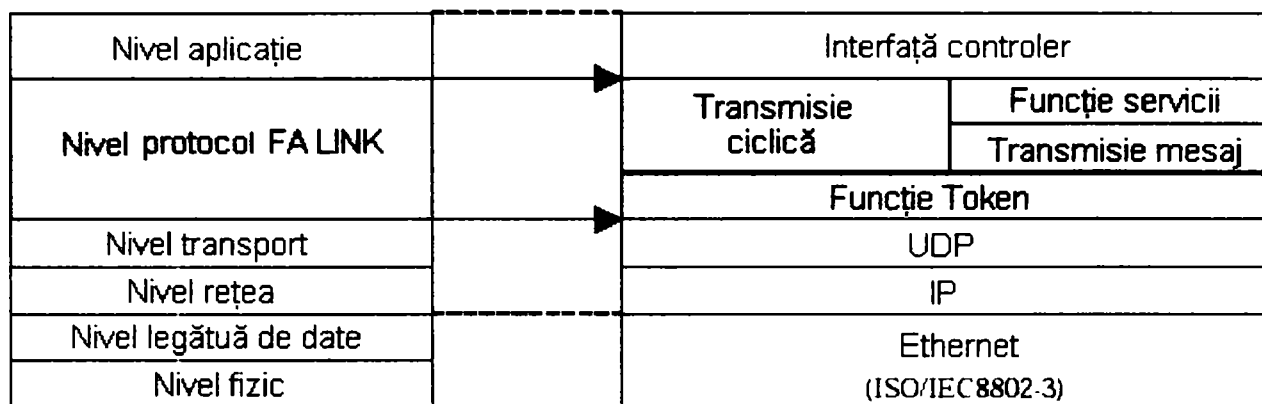


Fig.3.35. Arhitectura FL-net, [82]

Trebuie subliniat că **UDP** (User Datagram Protocol) asigură o viteză de transmisie mai mare a mesajelor mici, decât protocolul TCP (Transmission Control Protocol). Astfel, într-un mediu de control bazat pe mesaje scurte, această arhitectură poate asigura un control mai apropiat de timp real.

Între nivelul de aplicație și nivelul transport se găsește un nivel corespunzător protocolului numit "**FA LINK**". Acesta asigură între altele, dreptul fiecărei stații la transmisie de mesaje, prin utilizarea unui sistem cu jeton (token). Astfel, într-o perioadă de timp bine stabilită și relativ scurtă, fiecare stație intră în posesia jetonului și poate să transmită mesajele proprii. Sistemul este asemănător celui utilizat în cadrul structurilor MAP, numai că aici se utilizează pe nivelul 5 (în loc de nivelul 2). Astfel pentru primele două nivele rămâne posibilitatea utilizării unor soluții mai ieftine cum este Ethernet.

3.4.2. Introducerea unor structuri deschise de comandă destinate inițial domeniului proceselor industriale

O altă posibilitate pentru obținerea în cadrul Hipersistemelor CIM a unor modalități de comandă de la distanță, în variantă cu cost redus, este utilizarea unor structuri deschise, destinate domeniului proceselor industriale. Astfel de structuri sunt de exemplu: magistrale de câmp (fieldbus), DeviceNet, ControlNet, Ethernet/IP etc.

3.4.2.1. Structuri deschise de comandă cu magistrale de câmp

Structurile cu magistrale de "câmp" (fieldbus) au fost destinate interconectărilor informaționale în **domeniul proceselor industriale**, atât la nivelul automatelor

programabile (PLC), cât și la nivelul componentelor individuale de acționare respectiv de automatizare (în continuare: **dispozitive de câmp** – field device).

Scopul primordial introducerii magistrelor de câmp a fost **reducerea costurilor de cablare**, care apar la conectarea individuală la automate programabile, a dispozitivelor de câmp (cum sunt de exemplu senzorii și respectiv elementele de execuție – actuators). Soluția adoptată urmărește interconectarea inițială a acestora printr-o rețea LAN de tip “magistrală” și apoi, conectarea acesteia la automatul programabil printr-un singur cablu, fig.3.36.

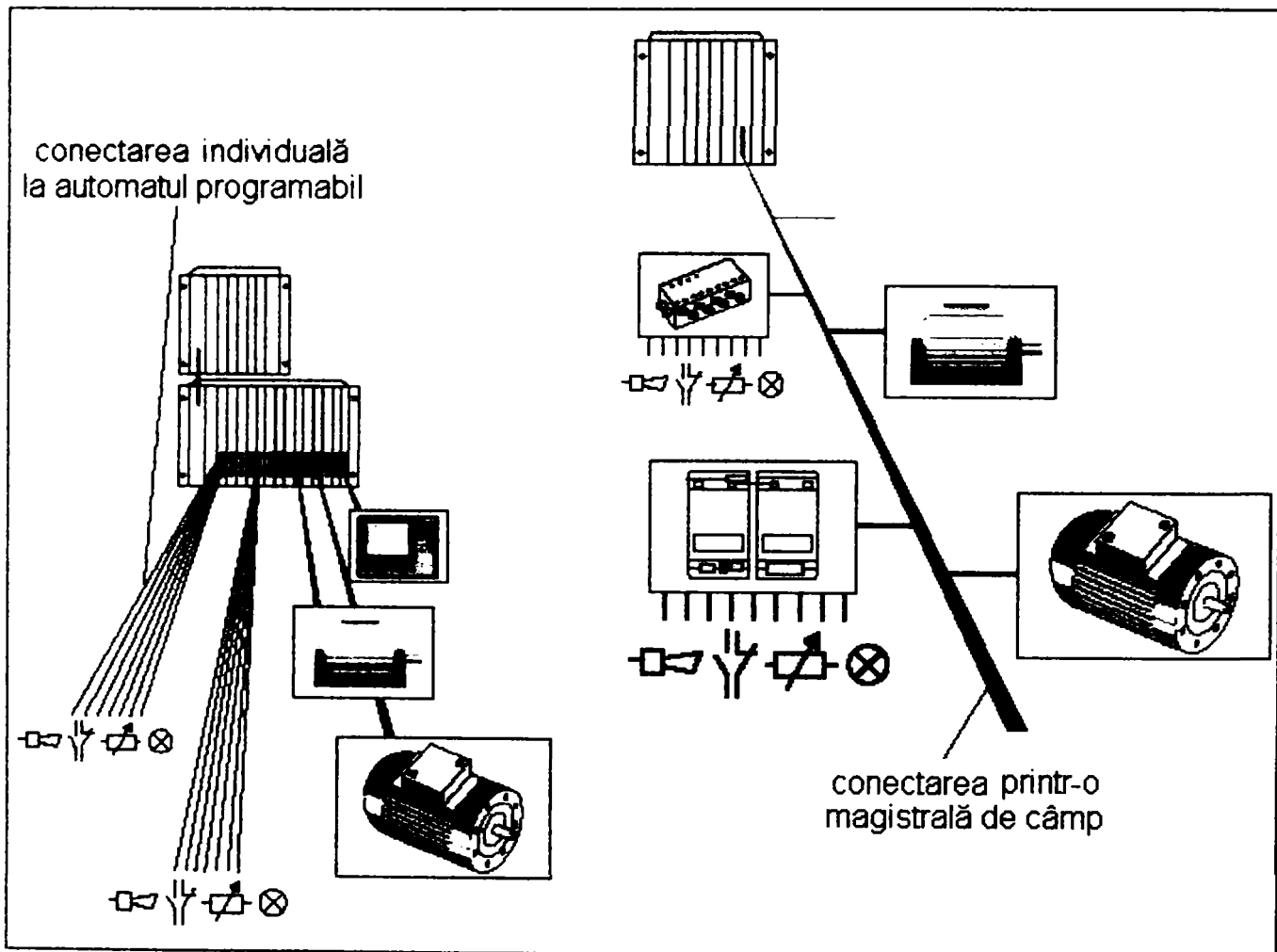


Fig.3.36. Principiul interconectărilor prin magistrală de câmp, [50]

Fiind promovat sub forma unor standarde deschise, se consideră că magistralele de câmp, treptat-treptat vor înlocui complet transmiterea tradițională, sub formă analogică a “informațiilor” (numită “**bucla de 4-20 mA**”), între automatul programabil (PLC) și senzori respectiv între PLC și elemente de execuție.

Magistralele de câmp oferă o metodă de **transmitere numerică** (“digitală”) a valorilor măsurate (de la senzori către PLC) respectiv a mărimilor de comandă (de la PLC către elementele de execuție), realizând astfel îmbunătățirea calității și creșterea eficienței controlului.

Există mai multe încercări pentru elaborarea unor specificații deschise referitoare la magistrale de câmp. Cele mai semnificative sunt: **Interbus** (standardizat în Germania – DIN 19258), **Foundation Fieldbus** (standardizat în Marea Britanie - DD 238) și **Profibus** (standardizat în Germania – DIN 19245).

Structuri Interbus

Structurile Interbus au fost elaborate în 1987 de firma Phoenix Contact. Inițial au fost destinate exclusiv realizării transmiterii datelor de proces, între automate programabile (PLC) respectiv senzori și actuatori, prin utilizarea unei interconectări printr-o singură magistrală.

În zilele noastre, Interbus oferă deja posibilitatea interconectării informaționale, pe lângă un număr mare de tipuri de dispozitive de câmp, respectiv interfețe operator de tip MMI (Man Machine Interface), și a unor echipamente de fabricație, cum sunt de exemplu roboții industriali, fig.3.37.

Prin introducerea modulelor de control (controller board) de generația a 4-a, există posibilitatea conectării structurilor Interbus și cu alte structuri deschise, cum este de exemplu Ethernet sub TCP/IP.

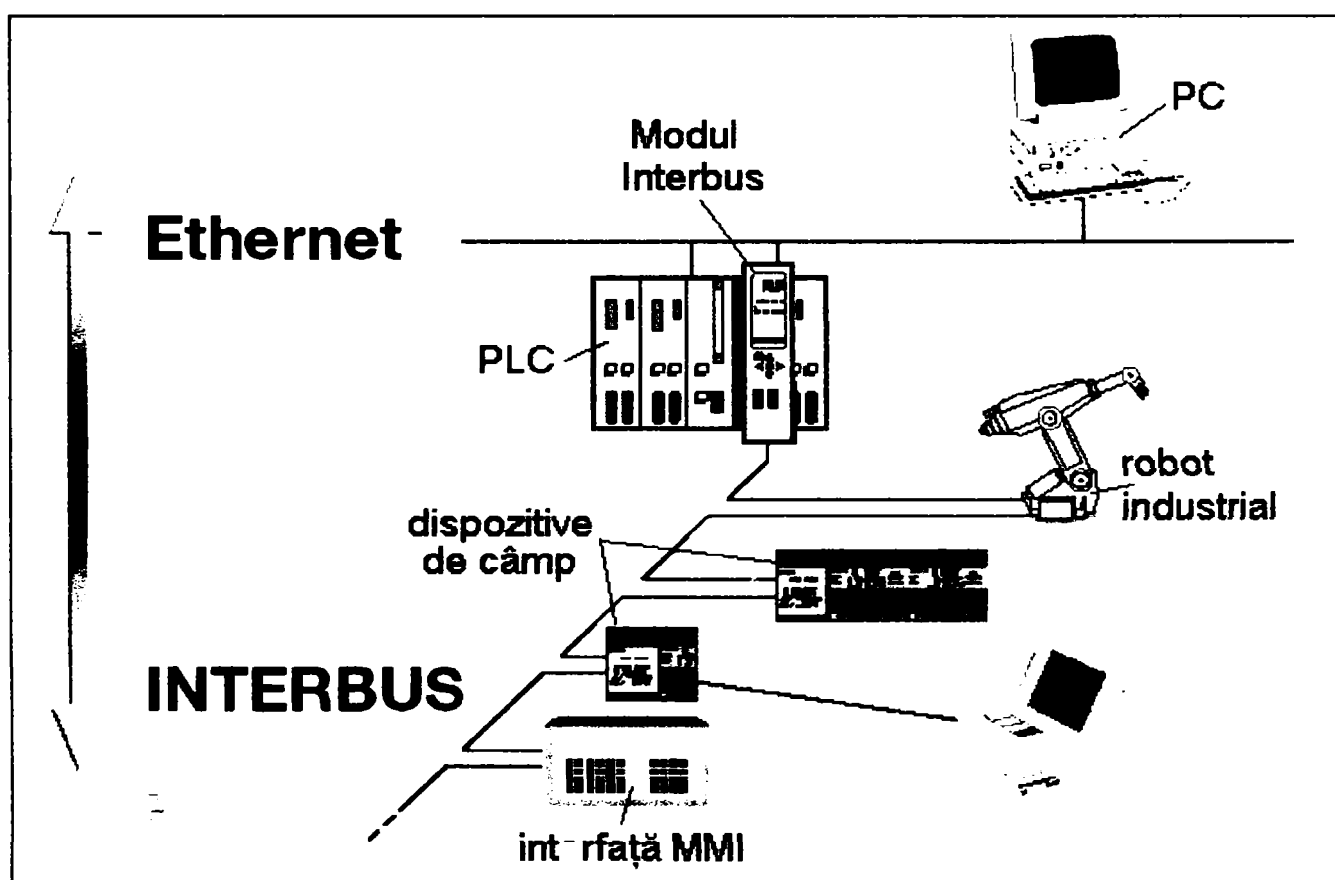


Fig.3.37. Structuri tipice de Interbus, [50]

Pentru transmiterea informațiilor, Interbus utilizează o arhitectură asemănătoare modelului de referință OSI. Însă pentru obținerea unei eficiențe cât mai bune, se utilizează doar 3 nivele (fizic, legătură de date și aplicație). La nivelul 7 sunt implementate două "canale" independente, aferente schimburilor de date, așa cum este reprezentată în fig.3.38.

Canalul de "date de proces" (proces data) servește la schimburi informaționale periodice (în timp) cu senzori și actuatori. Celălalt canal, a "datelor de parametru" (parameter data), servește la transmiterea informațiilor complexe, sub formă de mesaje, prin utilizarea unui model client-server.

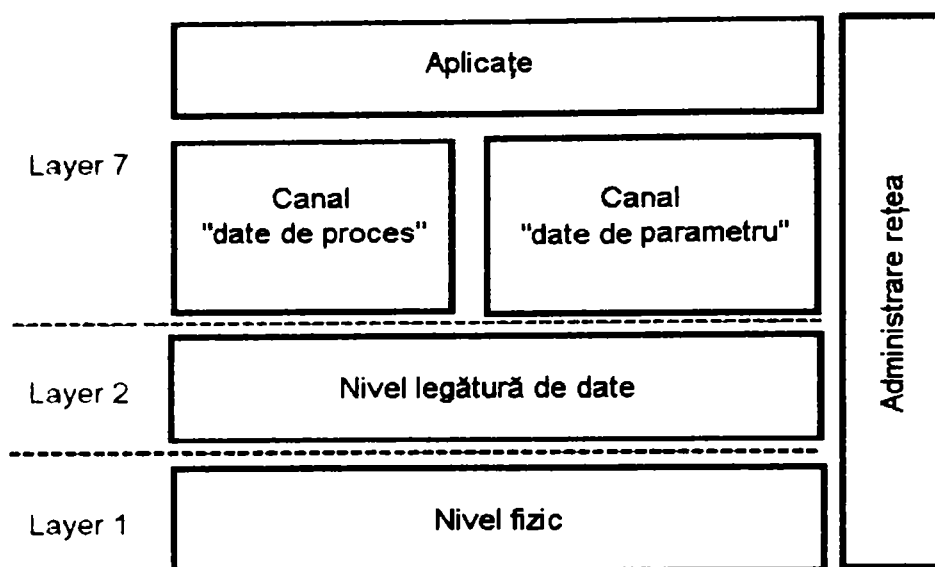


Fig.3.38. Arhitectura Interbus, [75]

Se consideră că Interbus oferă posibilitatea realizării unor structuri de control/comandă relativ simple și ieftine, cu aplicații atât în procese industriale cât și în sisteme de fabricație. Promovarea sub forma unui standard deschis, a condus la apariția a **peste 2000 de tipuri de dispozitive/echipamente compatibile**, care provin de la cca. 1000 de furnizori diferiți.

Structuri Foundation Fieldbus și Profibus

Foundation Fieldbus utilizează structuri asemănătoare celor de la Interbus. Diferența esențială este faptul că la nivel de aplicație este promovat un serviciu denumit **FMS** (Fieldbus Message Specification), fig.3.39.

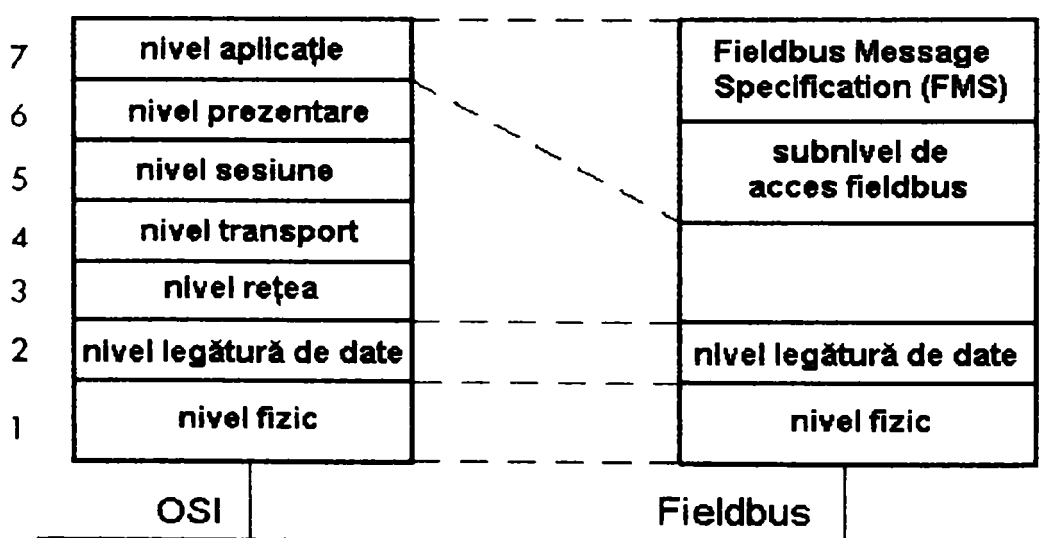


Fig.3.39. Arhitectura Foundation Fieldbus, [100]

FMS oferă realizarea unor schimburi informaționale, asemănătoare specificației MMS (Manufacturing Message Specification) din cadrul structurilor MAP. De asemenea utilizează o descriere abstractă a resurselor controlate (Virtual Field Device -VFD).

Structurile Profibus (prescurtare de la PROcess FieldBUS) au fost elaborate astfel încât să fie destinate atât proceselor industriale ("Process") cât și structurilor de fabricație ("Manufacturing"), fig.3.40.

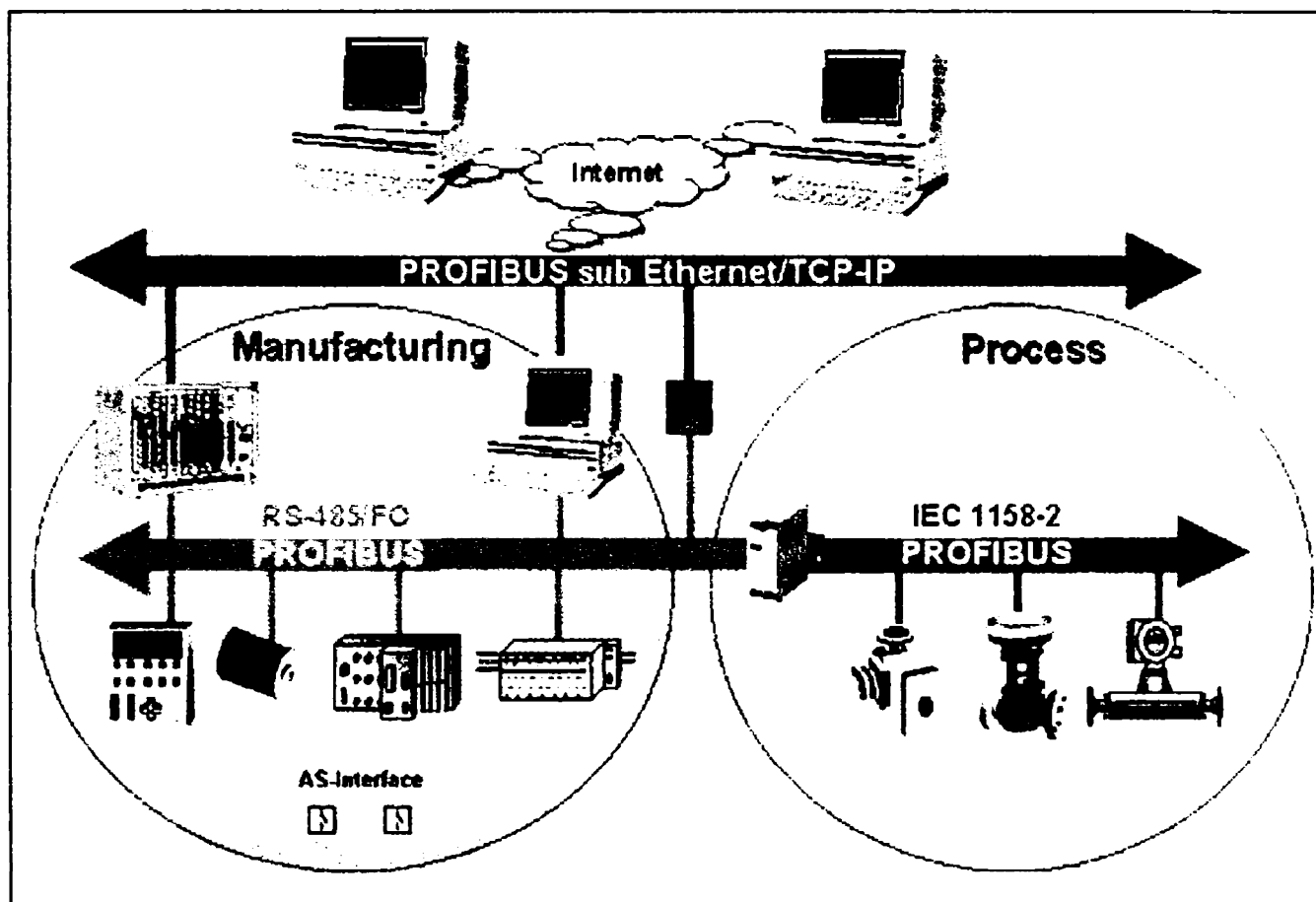


Fig.3.40. Profibus destinat atât structurilor de fabricație cât și proceselor industriale

Există mai multe variante de implementare (Profibus-FMS, Profibus-DP, Profibus-PA). La alegerea variantei optime se ține cont de natura procesului controlat. Pentru structurile de fabricație este recomandată varianta Profibus-FMS (având implementat FMS la nivel de aplicație). Aceasta poate fi utilizată pentru schimburile informaționale între echipamente de comandă și controler de celulă.

Având la bază un standard deschis, deja **peste 1400 de furnizori** oferă produse compatibile cu Profibus. A fost elaborată și o variantă sub Ethernet/TCP-IP.

3.4.2.2. Structuri deschise de comandă cu protocol "CIP"

O altă categorie de structuri de comandă, care de asemenea pot fi introduse în Hipersisteme CIM, sunt cele care utilizează un protocol comun numit **CIP (Control and Information Protocol)**. Acesta practic înglobează două protocoale de nivel de aplicație: unul de *Control* și unul de *Informație*. Primul este destinat efectuării în timp real a unui control de tip I/O (input/output), iar cel de-al doilea este destinat schimburilor de mesaje între echipamente și stații, [47]. Astfel CIP poate fi utilizat, în mod selectiv, la mai multe nivele de automatizare din cadrul unei întreprinderi. Astfel de nivele sunt: nivelul dispozitivelor; nivelul structurilor de comandă/control și nivelul integrărilor informaționale ale acestor structuri.

Pentru toate aceste trei nivele au fost dezvoltate structuri specifice care au la bază protocolul CIP. Astfel pentru nivelul dispozitivelor a fost dezvoltat sub denumirea de **DeviceNet**, pentru "nivelul de control" sub denumirea **ControlNet** iar pentru "nivelul informațional" sub denumirea **EtherNet/IP**. Legătura informațională între cele trei nivele este asigurată prin utilizarea Router-elor de CIP, fig.3.41.

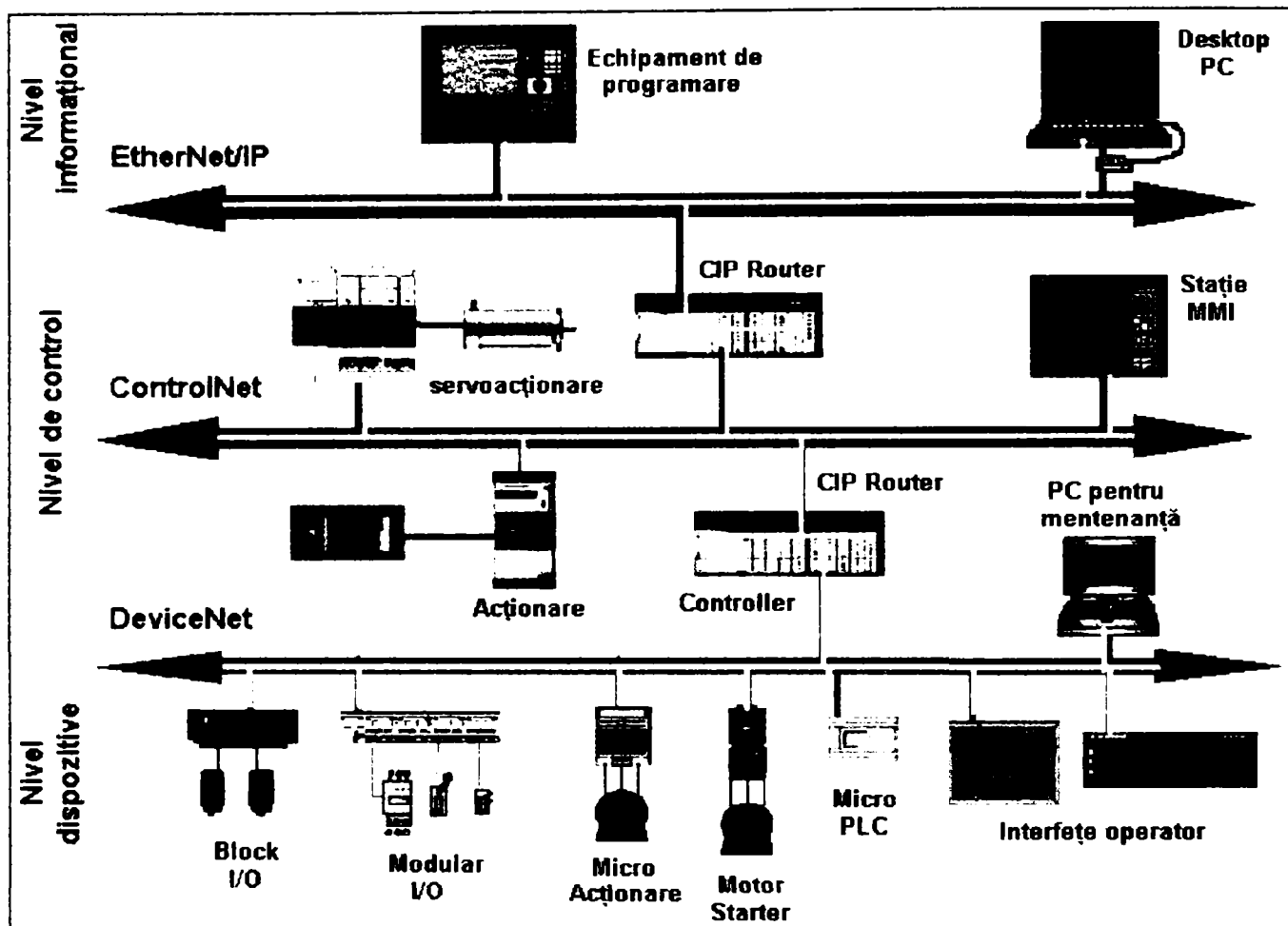


Fig.3.41. CIP în interconectarea nivelelor informațional, control și echipamente, [47]

Cele trei structuri, adică DeviceNet, ControlNet și EtherNet/IP se deosebesc doar la nivelele inferioare de comunicație (fizic, legătura de date etc.), pentru care s-au adoptat protocoale specifice aferente fluxurilor de date/informații, fig.3.42.

Trebuie remarcat faptul că DeviceNet utilizează la nivelul 2 un protocol deschis numit **CAN (Controller Area Network)**, care inițial a fost dezvoltat pentru comunicația din partea electronică a autoturismelor moderne, [63]. În prezent însă se utilizează și la multe echipamente de comandă din industrie ca interfață cu dispozitive (actuatori și senzori).

La nivel 2, DeviceNet utilizează un protocol numit **CTDMA (Concurrent Time Domain Multiple Access)**, [90]. Acesta are la bază un algoritm de divizare (între nodurile rețelei) a timpului de acces la canalul de transmisie. De asemenea este alocată o anumită prioritate, în funcție de natura datelor transmise. Astfel, acele date care sunt strâns legate de realizarea unui control în timp real, au prioritate mai mare.

EtherNet/IP utilizează structurile CSMA/CD (Ethernet) respectiv UDP/TCP/IP care au fost deja prezentate, [13].

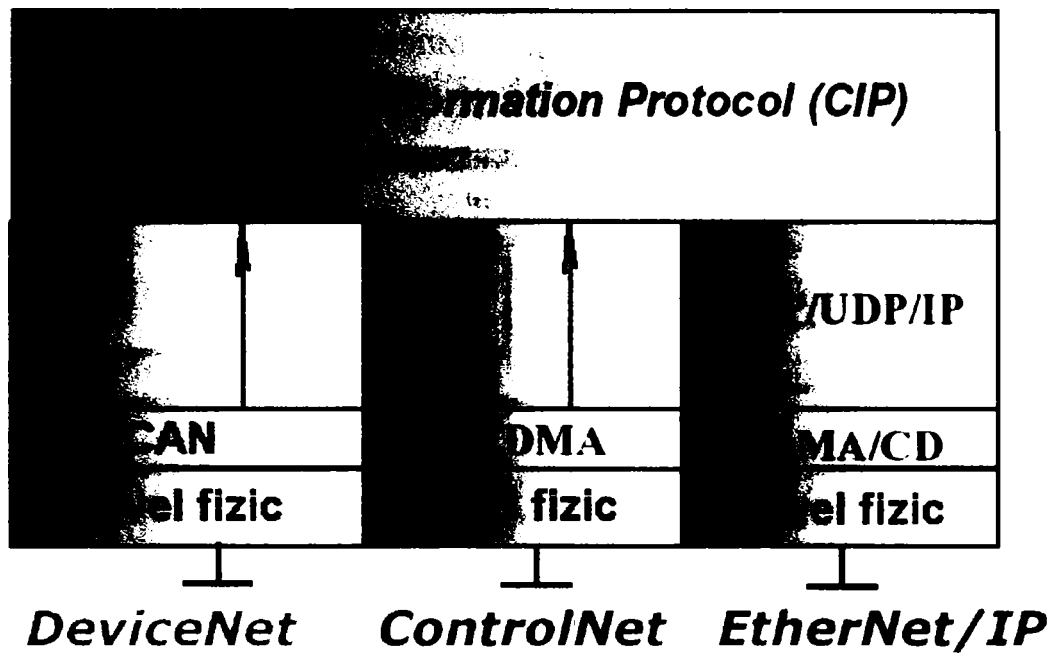


Fig.3.42. CIP protocol comun pentru DeviceNet, ControlNet și EtherNet/IP, [105]

În ultima perioadă, în centrul atenției sunt implementările cu EtherNet/IP, [67], care în structurile de fabricație ar putea înlocui protocoalele de pe nivelul de control, de exemplu protocolul ControlNet.

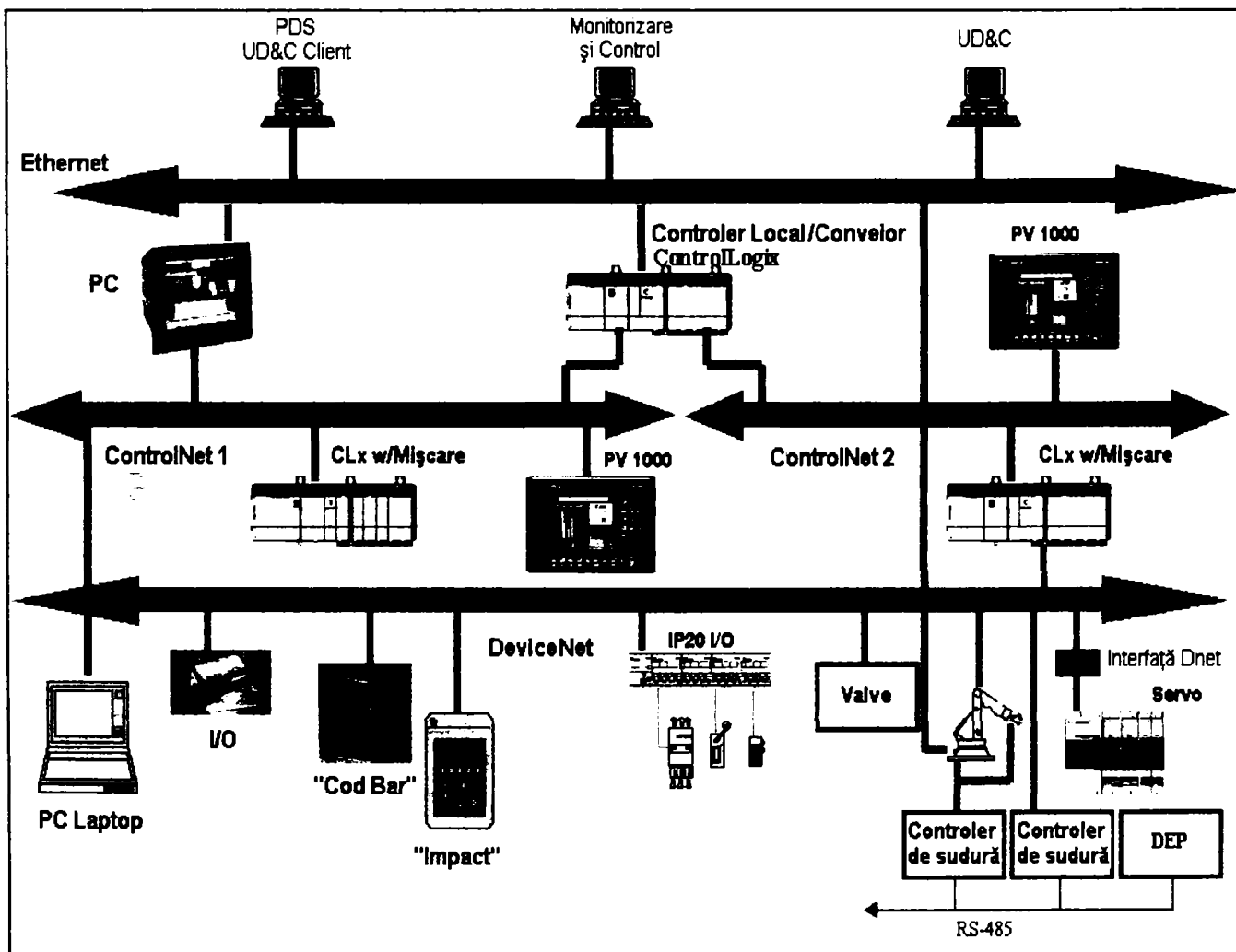


Fig.3.43. Situația inițială la General Motors, [109]

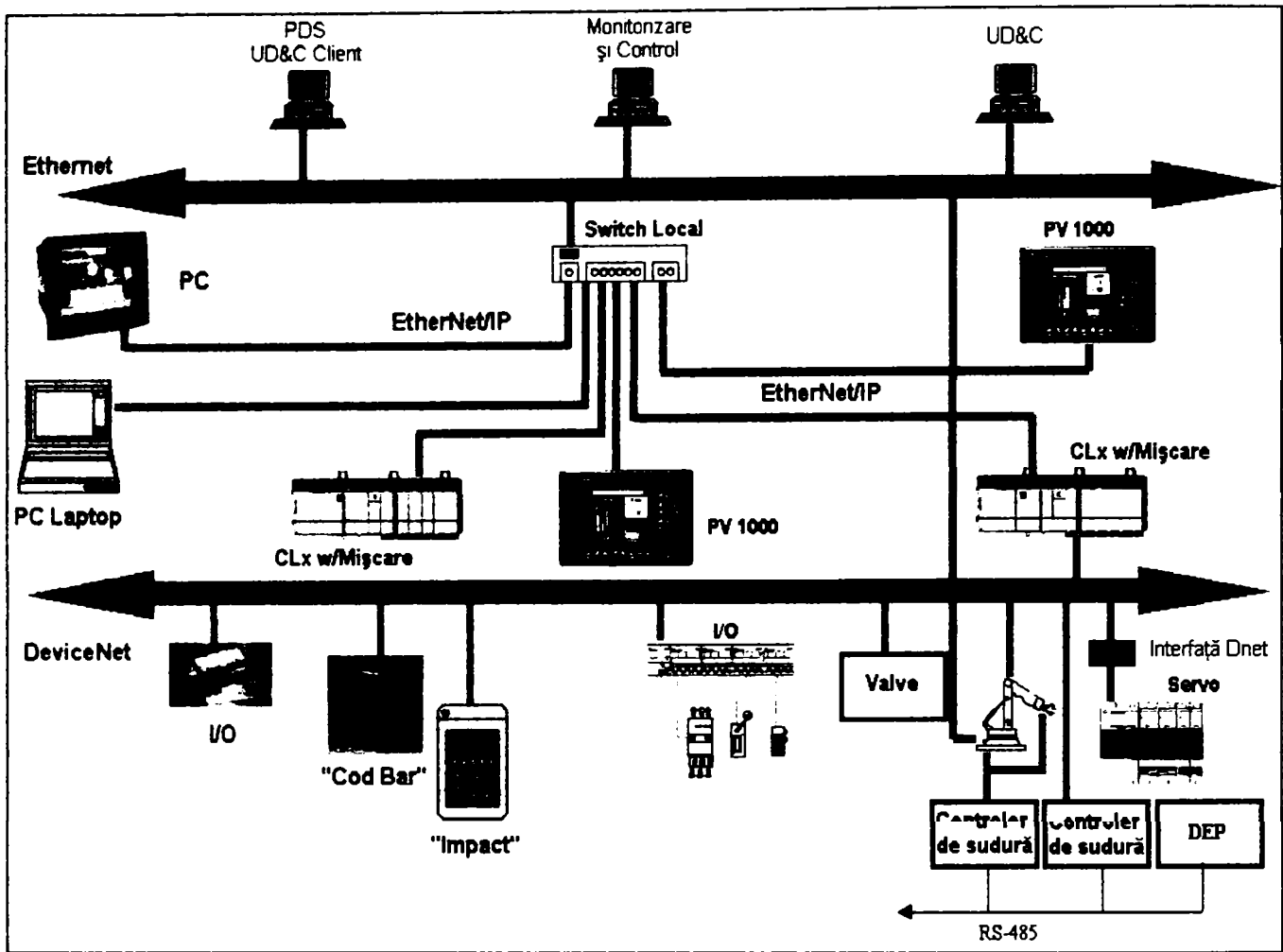


Fig.3.44. Faza pilot –I., [109]

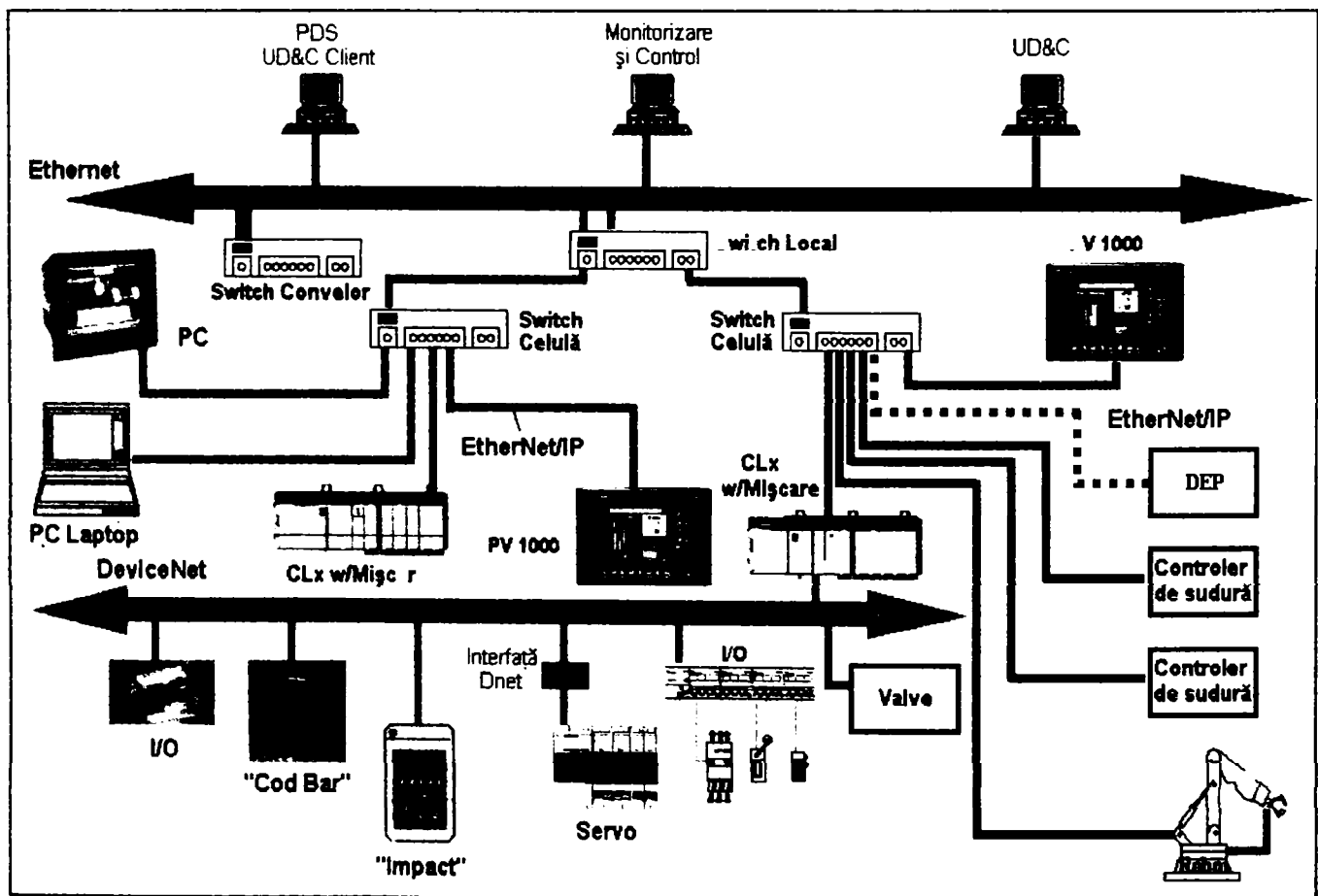


Fig.3.45. Faza pilot –II., [109]

Un exemplu destul de elocvent este **cazul firmei General Motors** care a decis printr-un proiect pilot, experimentarea introducerii structurilor Ethernet/IP, într-o secție de fabricație caroserii de automobile, [109]. Această secție utiliza inițial o structură de comandă organizată pe 3 nivele, cu DeviceNet, ControlNet și un protocol sub Ethernet +TCP/IP, fig.3.43.

Trebuie menționat faptul că roboții industriali respectiv controlerile “periferice”, cum este controlerul de sudură (weld controller), erau conectați la DeviceNet.

Pentru prima fază pilot s-a decis înlocuirea structurilor de ControlNet cu Ethernet/IP, fig.3.44. Aceasta era posibilă deoarece ambele structuri utilizează la nivel de aplicație același protocol comun: CIP. Pentru a 2-a fază pilot s-a prevăzut realizarea legăturilor cu Ethernet/IP până la nivelul roboților respectiv controlerelor periferice (controlere de sudură), fig.3.45.

3.5. Concluzii

În acest capitol s-a încercat analiza evoluției în cadrul Hipersistemelor CIM a structurilor deschise de comandă (“Open Control”). S-a pornit de la problema costurilor ridicate de integrare informațională, generate de incompatibilitatea informațională a echipamentelor de fabricație de proveniență diferită. S-a arătat că pentru reducerea costurilor de integrare există două modalități:

1. se achiziționează fiecare echipament de la același producător, care oferă în general compatibilitatea informațională între produsele proprii,
2. se utilizează în cadrul Hipersistemului CIM numai structuri deschise de comandă.

De asemenea s-a arătat că dintre cele două modalități enunțate numai cea de a doua este viabilă.

Condiția ca o structură de comandă să fie deschisă este ca specificația privind modul de realizare a comenzii să fie publică, adică orice utilizator sau fabricant să aibă acces gratuit la ea. Aceasta este condiția esențială pentru apariția pe piață a unui număr mare de echipamente de fabricație, provenite de la furnizori diferiți, care să fie compatibile între ele d.p.d.v informațional. Măsura gradului de deschidere al unei structuri de comandă este indicată totdeauna, de numărul produselor compatibile (dotate cu aceste structuri) de pe piața echipamentelor de fabricație, [65].

S-a arătat că primele structuri deschise de comandă, utilizate în cadrul Hipersistemelor CIM, au fost structurile MAP (Protocol de Automatizare a Fabricației). Cu aceste structuri puteau fi rezolvate toate problemele tehnice legate de automatizarea fabricației. Însă apărea o nouă problemă și anume prețul mai ridicat al echipamentelor de comandă, rezultate după dotarea lor cu interfețe MAP. Achiziționarea acestor echipamente a devenit neeconomică pentru majoritatea întreprinderilor mici și mijlocii, în care s-au încercat promovarea altor tipuri de structuri de comandă, cu cost redus.

Pentru introducerea în Hipersisteme CIM a unor structuri deschise de comandă cu cost redus s-a identificat două mari tendințe. Pe de o parte se încearcă intro-

ducerea unor protocoale ale "internetului", care în timp au devenit standarde "de facto" și astfel utilizarea lor este foarte ieftină. Pe de altă parte se încearcă introducerea în Hipersisteme CIM a unor structuri deschise de comandă, care inițial au fost destinate domeniului proceselor industriale.

Ambele tendințe oferă posibilitatea obținerii unor soluții economice, însă în unele cazuri ele se obțin în detrimentul performanțelor.

CAPITOLUL 4.

ANALIZA TENDINȚELOR DE DEZVOLTARE PENTRU HIPERSISTEME CIM A UNOR ECHIPAMENTE DE COMANDĂ CU ARHITECTURĂ DESCHISĂ ("OPEN CONTROLLER")

4.1. Necesitatea introducerii în Hipersisteme CIM a echipamentelor de comandă cu arhitectură deschisă

Dezvoltarea și introducerea în Hipersisteme CIM, a unor echipamente de comandă cu arhitectură deschisă (open controller), este o solicitare formulată de integratorii de sistem.

Observație:

- Integratorii de sistem sunt firme specializate în proiectarea, implementarea fizică și dezvoltarea structurilor de fabricație din cadrul Hipersistemelor CIM. În cazul unor comenzi din partea unei întreprinderi, ei achiziționează de la producători echipamentele de fabricație și după caz le integrează în structurile de fabricație existente sau creează noi structuri (de exemplu noi celule de fabricație) în cadrul întreprinderii respective. De asemenea ei realizează la cerere și adaptarea structurilor existente la noile cerințe ale procesului de fabricație.

În cazul echipamentelor de comandă (în continuare: **controlere**) cu arhitectură "închisă" (adică specifică firmei producătoare), integratorii de sistem au posibilități limitate în adaptarea acestora la cerințe noi ale procesului de fabricație. Dacă această adaptare, necesită dotarea controlerelor respective cu unele componente (interfețe) hardware și/sau software noi, integratorul de sistem este nevoit să apeleze la fabricantul controlerului. Acesta din urmă însă, efectuează modificările cerute numai dacă interfețele noi, rezultate în urma dezvoltării, vor aduce beneficii și fabricantului, adică ulterior vor putea fi vândute și la alți "clienți". În aceste condiții, integratorii de sistem solicită de la fabricanți **elaborarea controlerelor în variantă cu arhitectură deschisă** (open controller).

Trebuie menționat faptul că termenul de controler "deschis" nu se referă la un controler "universal", ci se referă la un controler care are la bază specificații cunoscute sau publicate, [37]. În schimb prin controler universal se înțelege un controler care poate fi utilizat pentru comanda mai multor tipuri de echipamente de fabricație. De fapt fiecare producător interpretează altfel acest termen, [65]. O definiție a unui controler deschis ar putea fi următorul:

"un controler cu hardware și sistem de operare standard respectiv cu specificații deschise pentru interfețe".

Observație:

- În domeniul sistemelor de calcul, calculatorul PC este un bun exemplu pentru un sistem cu arhitectură deschisă, fig.4.1. Având la bază calculatorul personal origi-

nal IBM, arhitectura hardware a PC-ului este considerată a fi un standard "de facto" în rândul calculatoarelor comerciale, [37]. Din acest motiv stă la baza realizării a multor tipuri de controlere deschise. La fel și sistemul de operare Windows, elaborat de firma Microsoft și utilizat în milioane calculatoare PC, este considerat a fi un standard "de facto".

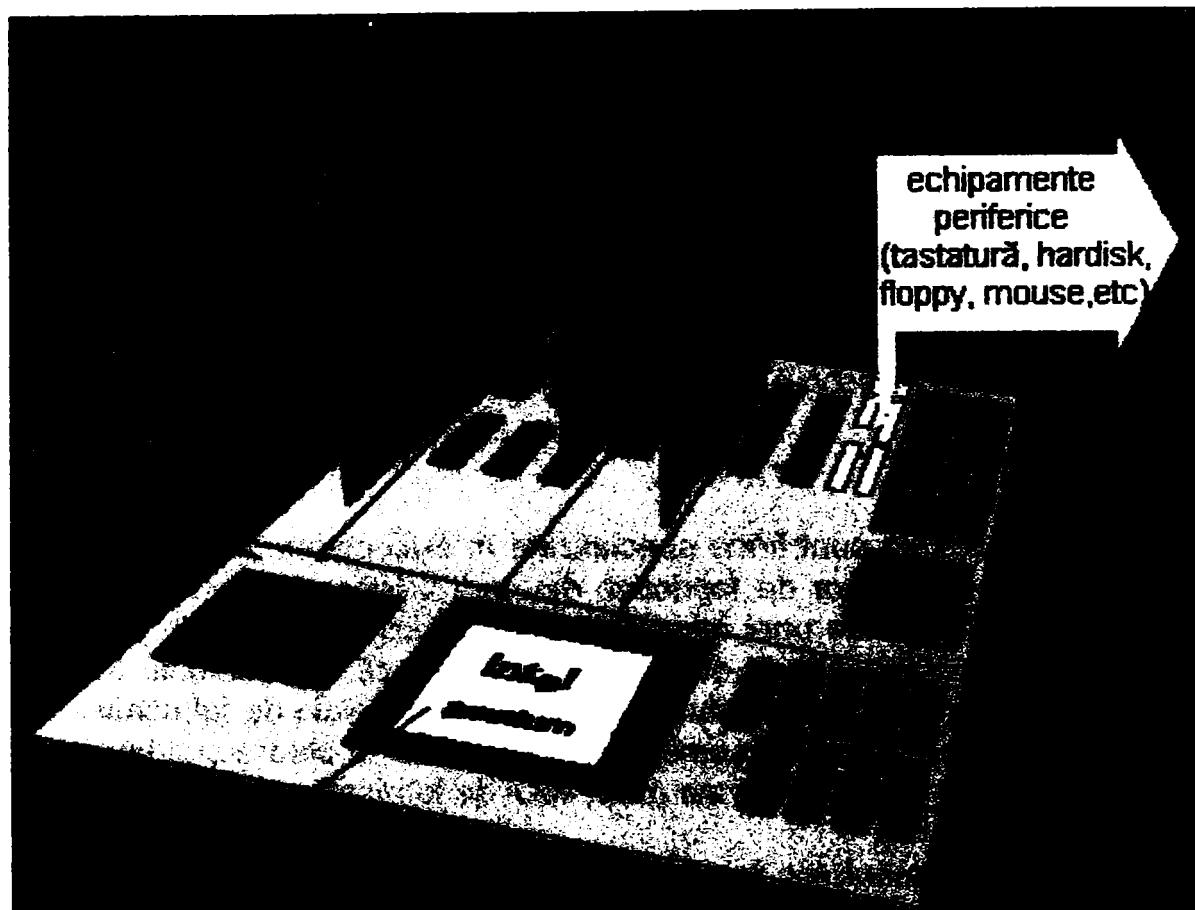


Fig.4.1. Tehnologia PC – un exemplu pentru o arhitectură deschisă

Controlerele brevetate (proprietary controllers), adică a căror arhitectură este specifică numai unei anumite firme producătoare, deși sunt permanent îmbunătățite, au o serie de dezavantaje față de un controler cu arhitectură deschisă. Astfel, majoritatea controlerelor brevetate sunt considerate a fi "insule de automatizare" datorită caracterului "închis" al acestora respectiv datorită compatibilităților și conectivităților reduse cu alte sisteme.

Multe controlere brevetate utilizează unu sau mai multe microprocesoare de uz general (de exemplu Intel 8088, Motorola 68000) dar specificațiile pentru restul hardware-ului și interfețelor au la bază standarde brevetate (respectiv sunt inaccesibile altora), [37]. Îmbunătățirea performanțelor hardware (prin introducerea unui procesor mai rapid sau prin extensia memoriei RAM), chiar dacă este posibilă totuși este limitată. Configurațiile interfețelor I/O sunt de asemenea standarde brevetate și contribuie la generarea problemelor de compatibilitate a sistemelor închise.

De exemplu, în cazul în care un controler utilizează o unitate floppy pentru înregistrarea datelor pe dischetă, dar formatul utilizat este unul brevetat (adică nu se utilizează formatul standard oferit de sistemul de operare DOS) atunci un operator va fi în imposibilitatea citirii datelor de pe aceasta, prin utilizarea unui simplu calculator PC.

În cazul utilizării în cadrul Hipersistemelor CIM a **controlerelor deschise**, toate aceste probleme pot fi rezolvate. Mai mult, hardware-ul și sistemul de operare fiind standard, în primul rând se creează posibilitatea ca mai mulți producători independenți, să ofere componente/module hardware și respectiv software pentru dezvoltarea lor prin "upgrade". Astfel va rezulta în timp (așa cum s-a întâmplat și în cazul calculatoarelor PC) o scădere semnificativă a prețurilor de vânzare. Și aceasta este un interes al integratorilor de sistem: adică **obținerea unor controlere cu preț scăzut** respectiv **posibilitatea dezvoltării și adaptării lor în funcție de necesitățile procesului de fabricație**. În final se obține o reducere a costurilor de integrare în Hipersisteme CIM.

4.2. Dezvoltarea echipamentelor de comandă robot (ROC) cu arhitectură deschisă

Controlerele robot fac parte din rândul acelor echipamente de comandă a căror arhitectură deschisă contribuie la obținerea unor costuri de integrare redusă în Hipersisteme CIM.

În prezent există mai multe variante de structură, pentru controlere robot cu arhitectură deschisă. În fig.4.2. s-a reprezentat o anumită variantă, în care sistemul mecanic, sistemul de acționare, intrările și ieșirile (I/O) respectiv panoul de instruire sunt considerate componente care au la bază standarde de firmă (standarde brevete), [79]. Sistemul de comandă și hardware-ul controlerului de comunicație respectiv specificațiile aferente sunt componentele deschise ale arhitecturii.

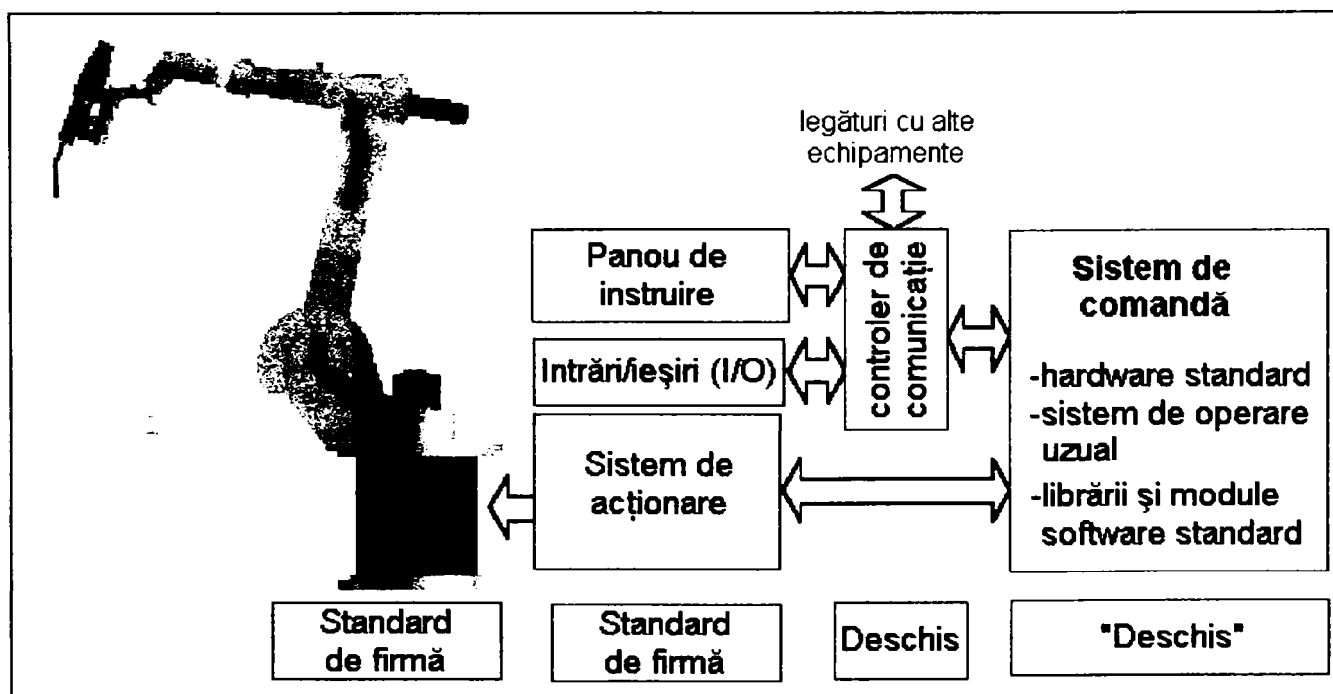


Fig.4.2. Exemplu de diagramă bloc a unui controler robot cu arhitectură deschisă

Eticheta de controler "deschis" ("open" controler) se referă în general la arhitectura deschisă a sistemului de comandă, [14]. Acesta din urmă este considerat deschis dacă este dotat cu un hardware standard, utilizează un sistem de operare uzual și permite accesul la software-ul intern de comandă printr-o serie de librării și module standarde, [79].

4.2.1. Evoluția echipamentelor de comandă robot către o arhitectură deschisă

În funcție de tipul arhitecturii utilizate, controlerul robot moderne pot fi grupate în 5 mari categorii, [25]:

1. Controler cu hardware și sistem de operare brevetat;
2. Controler cu hardware de PC și sistem de operare brevetat;
3. Controler cu hardware de PC și sistem de operare Windows95 și VxWorks într-o soluție dubluprocesor;
4. Controler cu hardware de PC și hardware brevetat, cu sistem de operare specific și adițional Windows95 într-o soluție dubluprocesor;
5. Controler cu hardware de PC cu sistem de operare Windows95 și VxWorks într-o soluție monoprosesor.

Cele mai "închise" sunt acele controlere de robot, care au la bază un **hardware și un software de sistem de operare brevetat** (categoria 1). Ele sunt caracterizate prin faptul că atât hardware-ul precum și software-ul utilizat pentru conducere (comandă și acționare) au la bază specificații închise, care sunt utilizate numai de un anumit fabricant. În majoritatea cazurilor ele sunt considerate secrete de firmă. În aceste condiții orice modificare hardware sau software a controlerului, în vederea dezvoltării sau adaptării, poate fi făcută numai de firma producătoare.

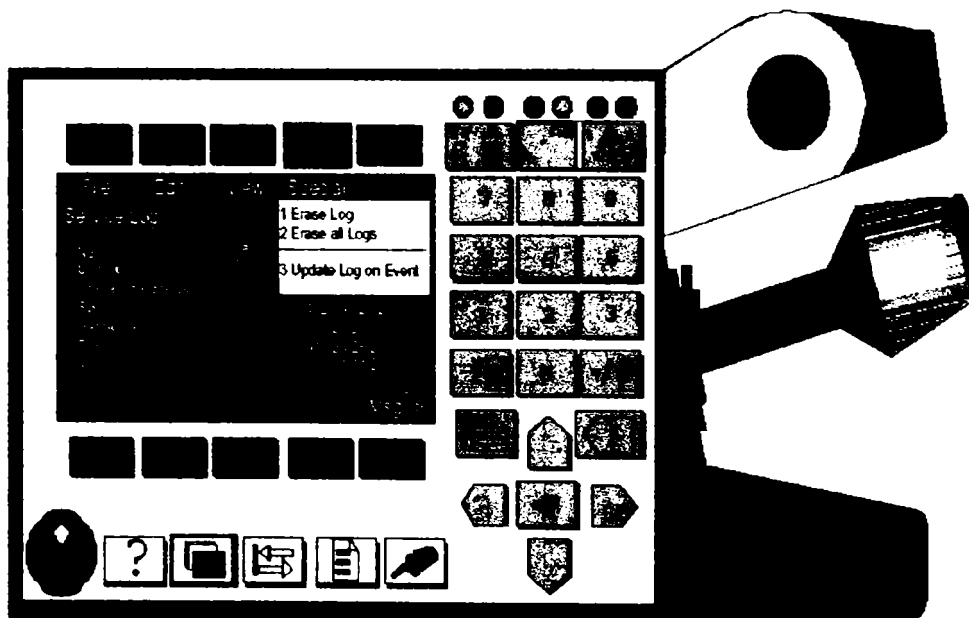


Fig.4.3. Exemplu de controler de robot cu hardware și sistem de operare brevetat

La aceste controlere, protocolul de comunicație utilizat la legătura cu alte echipamente, este în general deschis. Astfel ele permit conectarea la un calculator PC adițional, pe care este rulat un sistem de operare deschis (de exemplu Windows95/NT). Pentru acest calculator PC pot fi realizate, o serie de pachete software pentru programarea limitată a aplicațiilor, în mod text sau în mod grafic. Însă accesul la software-ul intern, legat de modul de realizare a conducerii, este imposibil pentru un utilizator. În general display-ul controlerului de robot nu are capacități grafice, fig.4.3., [25].

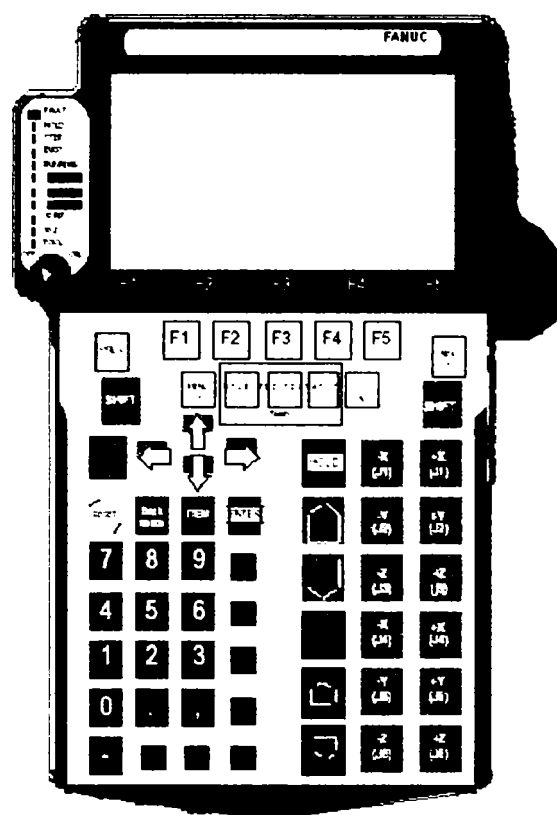


Fig.4.4. Exemplu de panou de programare cu display cu grafică limitată, [25]

În a doua categorie intră acele controlere care au la bază un **hardware standard de PC și un software de sistem de operare brevetat** (categoria 2). Trebuie subliniat că foarte mulți utilizatori agreează idea utilizării controlerelor deschise bazate pe tehnologia PC, [38]. Motivul este performanțele ridicate ale microprocesoarelor și faptul că prețurile chip-urilor, respectiv ale interfețelor de conectare la rețea, sunt relativ mici.

Aceste controlere însă, deși hardware-ul de PC oferă o platformă deschisă pentru realizarea conducerii, datorită software-ului brevetat, capătă un caracter închis. Ca avantaj, hardware-ul de PC asigură un preț mai scăzut pentru controler, respectiv asigură posibilitatea unui "upgrade" cu componente standarde, [38]. La aceste controlere, în general există o anumită capacitate grafică a panoului de programare, însă aceasta este restricționată de sistemul de operare care este puțin dezvoltat în acest sens, fig.4.4.

Pentru a avea o platformă de operare într-adevăr deschisă, la controlerele noi a fost introdus, pe o arhitectura de PC, sistemul de operare Windows95/98. Însă trebuie menționat faptul că Windows95/98 nu este sistemul de operare cel mai potrivit pentru realizarea controlului în timp real. În schimb oferă o capacitate ridicată pentru integrarea fără probleme a unor module hardware și software adiționale.

Observație:

- A fost dezvoltată și o variantă de "WindowsNT" care oferă o capacitate limitată pentru realizarea unui control în timp real, însă abilitatea de a integra hardware și software adițional este foarte redusă. O altă încercare (dar cu rezultate relativ slabe), a fost elaborarea sistemului de operare Windows2000, care încerca înglobarea tehnologiei NT într-o platformă deschisă (oferită de variantele 95 și 98).

În aceste condiții a fost promovată o soluție cu **două sisteme de operare** rulate împreună, [99]: un sistem de operare în timp real și un sistem de operare care oferă o platformă deschisă: Windows95 sau 98.

Un adevărat sistem de operare în timp real este **VxWorks** realizat de firma Wind River System. Acesta din urmă este utilizat alături de sistemul de operare Windows95/98 în două variante:

- **cu dublu procesor**, caracteristic controlerelor din categoria 3 și
- **cu monoprosesor**, caracteristic controlerelor din categoria 5.

În varianta **cu dublu procesor** (categoria 3), se utilizează practic o arhitectură hardware de PC **cu două plăci de bază**, fiecare fiind echipată cu câte un microprocesor. Astfel sistemele de operare Windows95/98 și VxWorks sunt rulate separat, pe câte o placă de bază care însă comunică între ele prin protocolul TCP/IP sub Ethernet, fig.4.5.

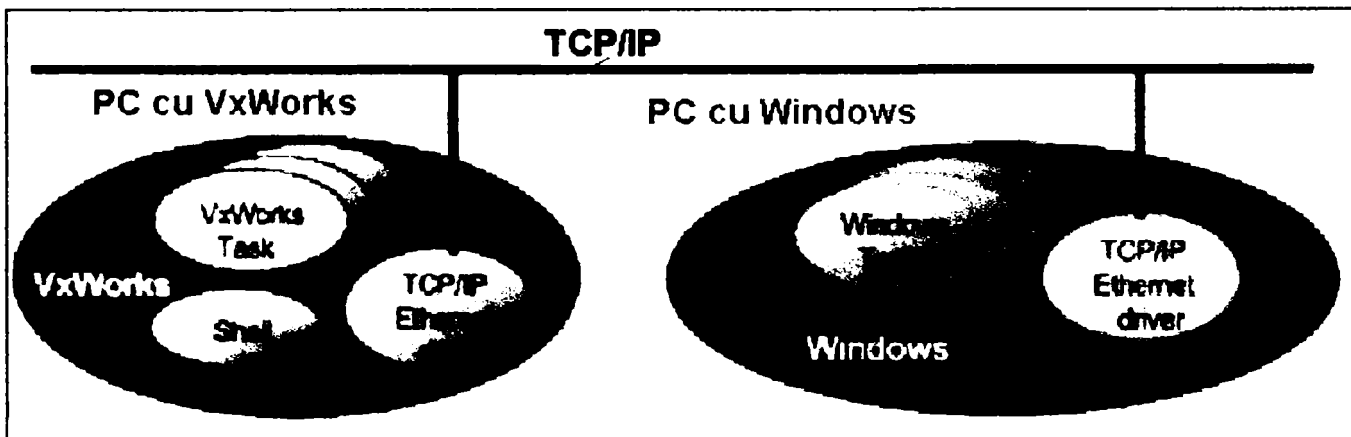


Fig.4.5. Comunicația tipică între VxWorks și Windows în varianta biprosesor, [48,83]

VxWorks este utilizat pentru prelucrarea tuturor task-urilor de timp real (de exemplu generarea traiectoriei etc.) iar Windows95/98 este utilizat ca bază a funcțiilor de operare și programare, [48]. Aceste funcții sunt realizate prin utilizarea unei tastaturi și a unui monitor standard. La aceste controlere nu este utilizat separat un panou de programare. De asemenea, controlerul în general este oferit fără o interfață utilizator concretă. În schimb fabricantul furnizează o serie de librării și module software deschise pentru sistemul de operare Windows, care pot fi utilizate de integratorii de sistem pentru crearea unor interfețe utilizator adecvate aplicației.

Având în vedere că într-o soluție biprosesor, platforma deschisă este oferită doar de sistemul de operare Windows, pentru partea de control în timp real în esență poate fi folosit și un hardware respectiv sistem de operare brevetat. Aceasta este caracteristică **controlerelor din categoria 4**. Aceste controlere deși oferă o capacitate grafică ridicată, legată de operare și programare, totuși utilizează pentru conectarea la alte echipamente respectiv pentru integrarea în Hipersisteme interfețe speciale, nestandarde.

O soluție ideală, d.p.d.v. al prețului de cost, este implementarea celor două sisteme de operare, de exemplu **VxWorks și Windows, pe un singur hardware de PC** (categoria 5), adică într-o soluție monoprosesor.

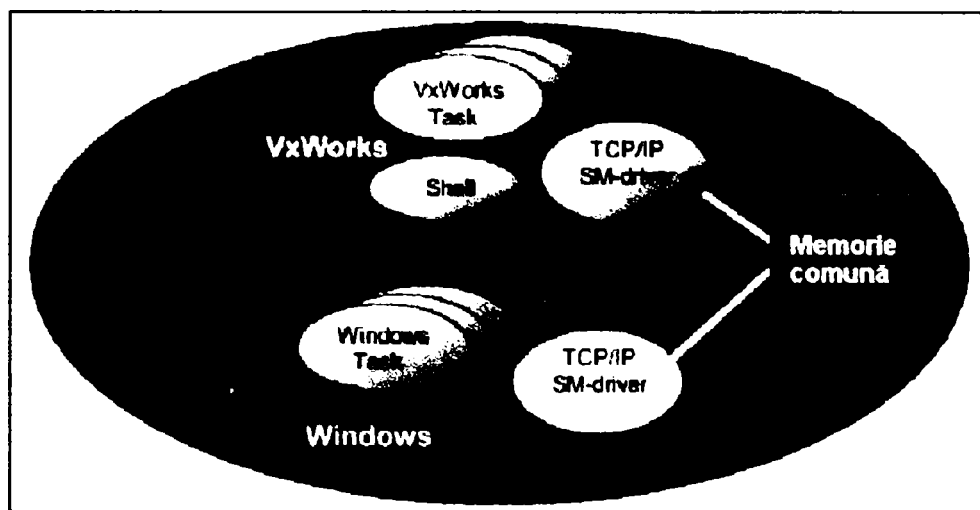


Fig.4.6. Comunicația tipică între VxWorks și Windows în varianta monoprosesor, [48]

Practic pe aceeași platformă PC (adică prin utilizarea unui singur microprocesor), sunt rulate simultan două sisteme de operare. Ele comunică prin intermediul unui "driver" TCP/IP utilizând o zonă de memorie comună (shared memory). Pentru garantarea unui control în timp real, VxWorks are prioritate de execuție față de Windows. Practic când un task de VxWorks este activ (de exemplu un task legat de generarea traiectoriei) timpul de execuție procesor este disponibil doar pentru VxWorks. Cu alte cuvinte, task-urile Windows-ului sunt executate numai în acele perioade de timp, când nici un task VxWorks nu este activ.

Aceste controlere posedă toate avantajele tehnologiei PC. Oricând permit integrarea unui hardware adițional prin utilizarea magistralei ISA/PCI. De asemenea oferă posibilitatea utilizării interfețelor standarde de conectare la rețea.

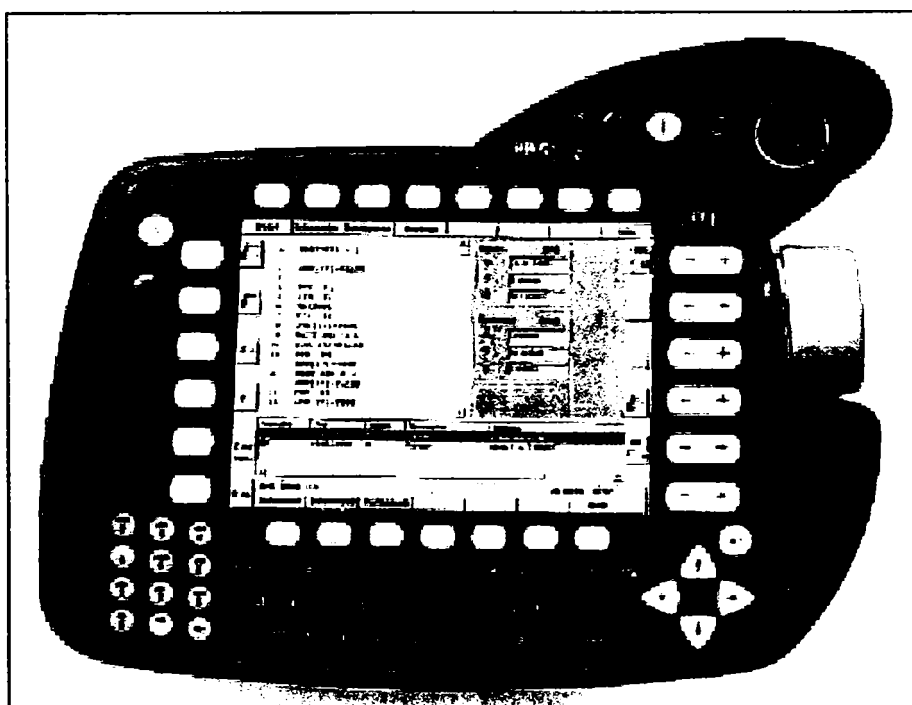


Fig.4.7. Panoul de programare al controlerului KUKA KR C1, [74]

Un exemplu pentru o arhitectură cu VxWorks și Windows pe o singură platformă PC, este controlerul KR C1 al robotului KUKA, fig.4.7. În fig.4.8. s-a încercat reprezentarea evoluției controlerelor de robot către o arhitectură deschisă.

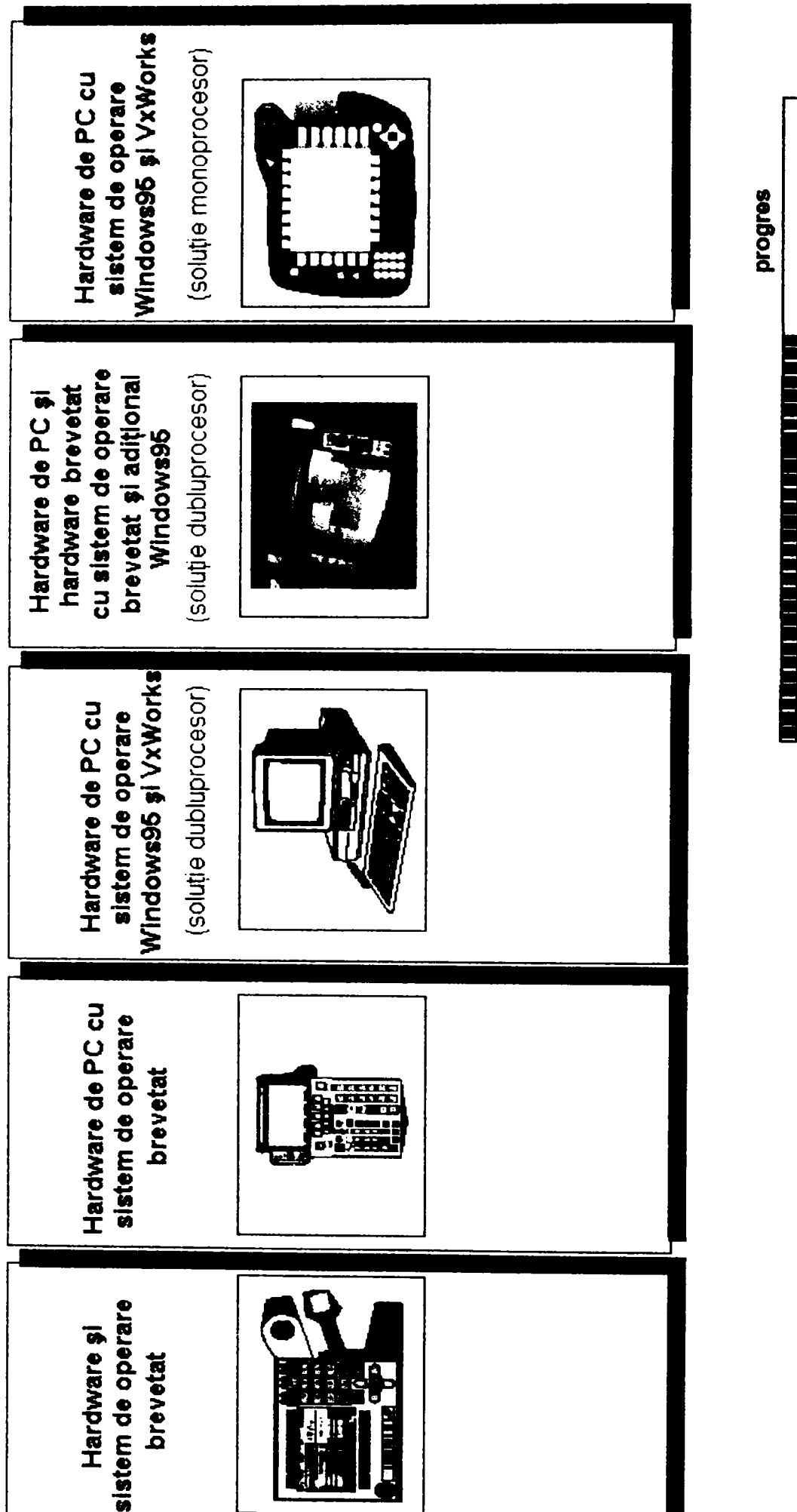


Fig.4.8. Evoluția controlerelor de robot către o arhitectură deschisă, [25]

4.2.2. Exemple de echipamente de comandă robot cu arhitectură deschisă

În continuare se prezintă două exemple pentru controlere de robot cu arhitectură deschisă: un controler Bosch IQ200 al robotului Turboscara SR6/SR7 și un controler KR C1 al unui robot KUKA.

4.2.2.1. Controler Bosch IQ200 al robotului Turboscara SR6/SR7

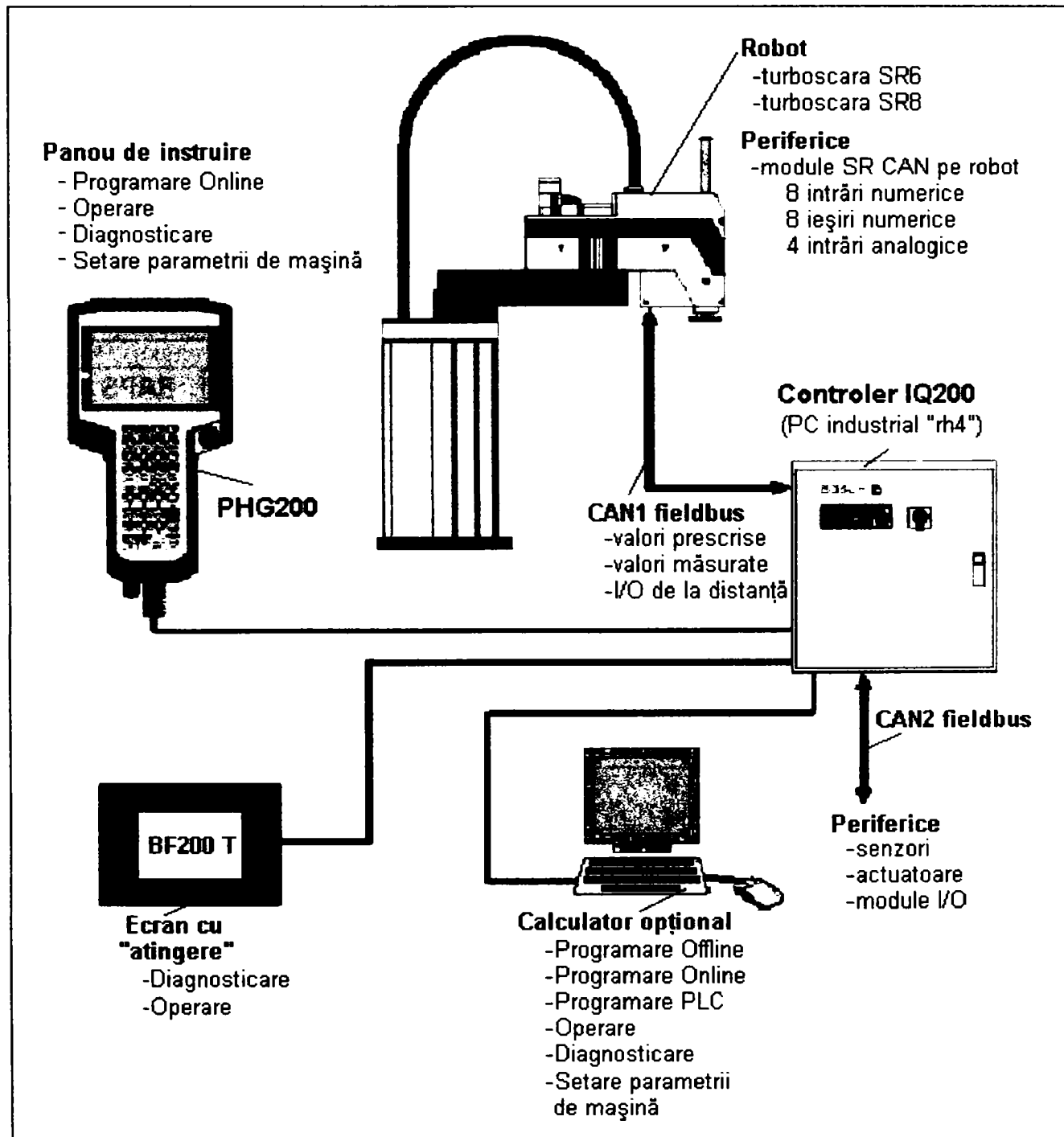


Fig.4.9. Sistemul robot Bosch – Turboscara, [48]

Bosch IQ200 este un exemplu tipic pentru un controler robot cu arhitectură deschisă. Este utilizat la familia de roboți TurboScara SR6/SR8. La baza controlerului

lui stă un PC industrial numit "rh04", cu un procesor Pentium MMX. Pentru conectarea la controler a modulelor de conducere nemijlocită ale celor 4 axe de robot se utilizează o magistrală de câmp (fieldbus) de tip CAN. O altă magistrală (CAN2) este utilizată pentru conectarea la controler a dispozitivelor periferice: senzori, actuatori, module I/O etc. Pentru operare, poate fi utilizat panoul de instruire PHG2000, ecranul de "atingere" (touchscreen), BF200T sau un calculator PC extern.

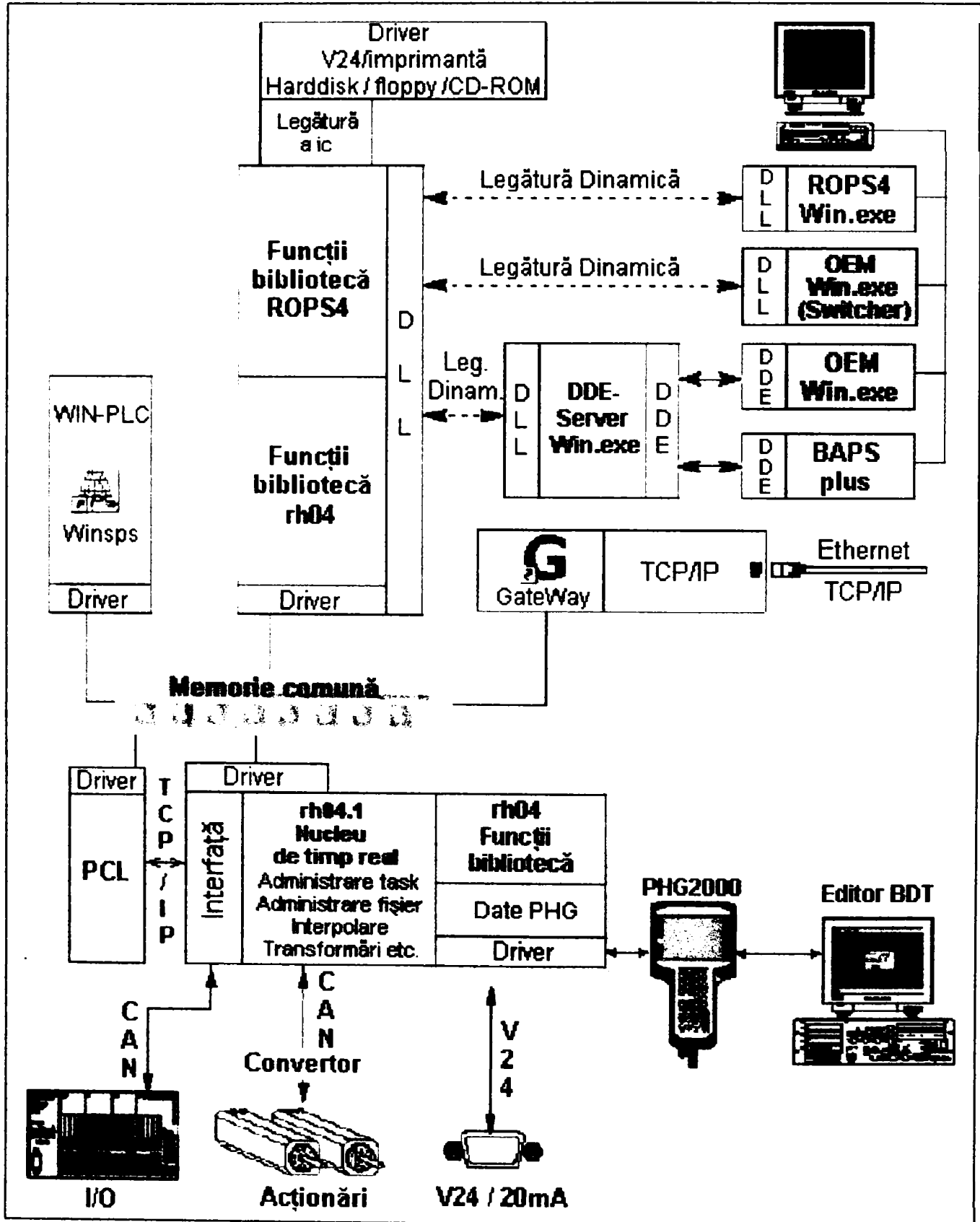


Fig.4.10. Arhitectura calculatorului industrial "rh04" din cadrul controlerului IQ200

Pe calculatorul rh04 este rulat simultan sistemul de operare de timp real **VxWorks** și sistemul de operare **Windows95** sau **NT**. Pe magistrala PCI este instalat un modul de extensie denumit PCI_RHO, fig.4.10. Acesta include: interfețe pentru magistrala CAN către acționări (cele 4 axe ale robotului) și către dispozitivele periferice (senzori, actuatori) respectiv o interfață către panoul de instruire PHG2000.

O altă funcție cheie a calculatorului rh04 este utilizarea protocolului TCP/IP pentru conectarea la o rețea standardă. Aceasta permite atât schimburile de date între modulele software interne cât și comunicația externă cu aplicații standard sau aplicații utilizator. De asemenea este implementată o capacitate de diagnosticare de la distanță via TCP/IP.

Prin utilizarea funcțiilor noi de bibliotecă (library) este posibilă **inițierea mișcărilor la axe, de la o aplicație Windows**. Pentru programarea robotului poate fi utilizat calculatorul industrial rh04 din interiorul controlerului IQ200 sau un calculator PC extern. În acest caz din urmă, datele și/sau programele pot fi introduse în memoria controlerului printr-o interfață serială sau printr-o legătură Ethernet sub TCP/IP. Panoul de instruire este utilizat în prima fază pentru instruirea/învățarea robotului și pentru diagnosticarea erorilor. De asemenea permite și modificarea sau editarea programului robot de aplicație.

Programele de “mișcare” robot sunt scrise într-un limbaj asemănător PASCAL-ului numit BAPS. Programele robot BAPS pot fi create cu un editor de texte sau cu un utilitar de programare grafică: BAPSplus. Pentru realizarea funcțiilor de automat programabil (PLC) se utilizează un software WIN-PLC.

În cadrul calculatorului industrial rh04, se utilizează un procesor Pentium de mare performanță. Acest microprocesor este **secționat în două procesoare virtuale P1 și P2**. **Procesorul P1** este procesorul de limbaj BAPS. Interpretează comenzile și planifică desfășurarea acțiunilor și traiectoria pentru mișcare (calculează în avans până la 11 “puncte”). În mod adițional, procesorul P1 este responsabil de legătura cu sisteme vision respectiv cu calculatoare Host, asigură funcțiile PLC respectiv asigură realizarea schimburilor de informații cu un operator, printr-o interfață Windows.

Task-ul procesorului P2 este generarea punctelor prescrise, separat pentru fiecare axă, respectiv realizarea transformărilor de coordonate. Monitorizează limitele spațiului de lucru sau limitele de cursă pentru axe.

De fapt calculatorul rh04 poate controla simultan până la 16 axe. Funcțiile extinse, cum sunt interconectarea cu ale sisteme prin TCP/IP, comunicația cu operatorul prin interfețe standarde DDE și DLL și utilizarea tehnologiei PC pentru realizarea conducerii, asigură o veritabilă arhitectură deschisă.

4.2.2.2. Controler KR C1 al unui robot KUKA

Controlerul KR C1 este realizat de firma KUKA Roboter GmbH. Are la bază un hardware de PC cu sistem de operare Windows 95 (realizat de firma Microsoft) și VxWorks (sub formă de LP-VxWin realizat de LP Elektronik GmbH), într-o variantă monoprosesor.

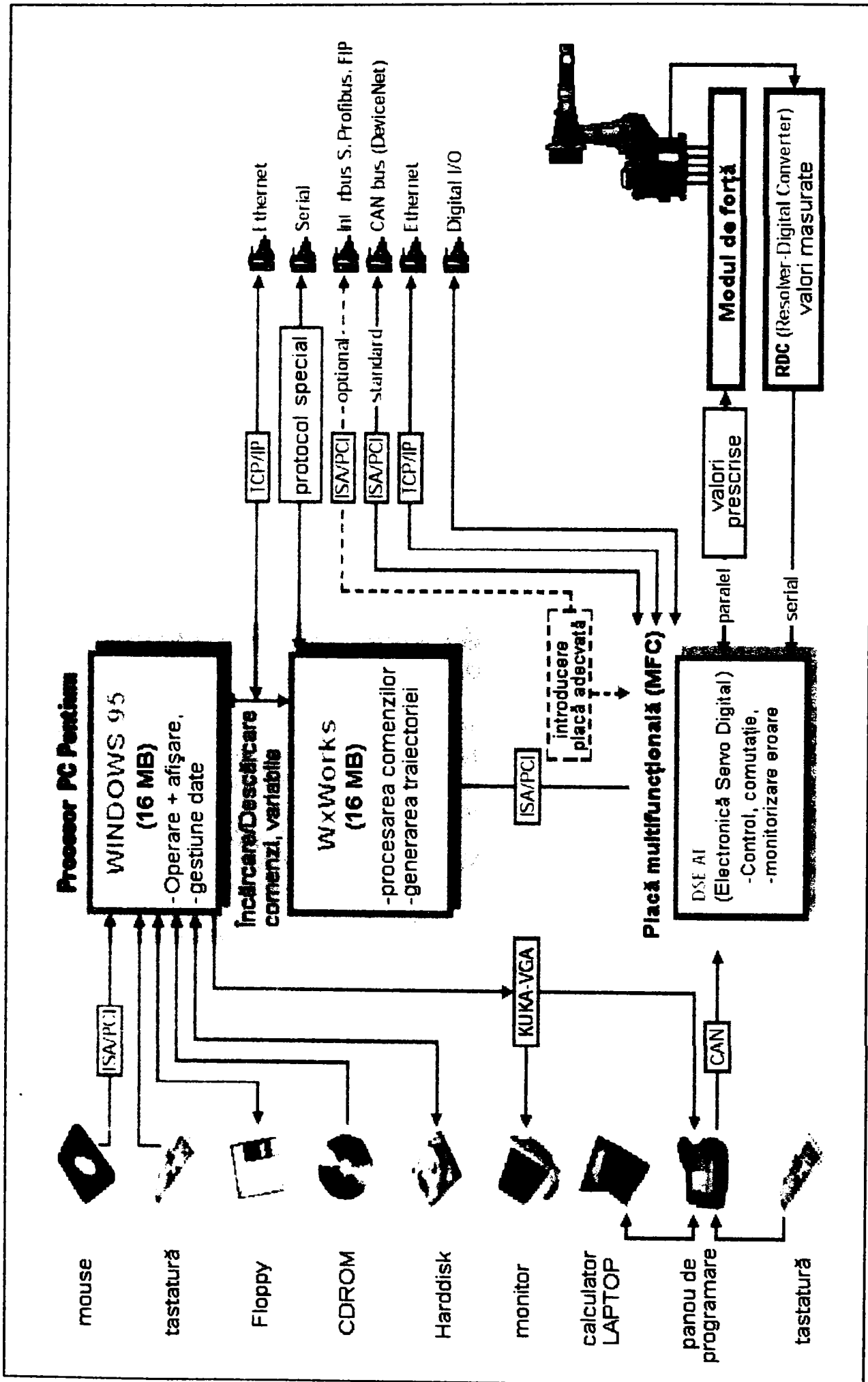


Fig.4.11. Arhitectura controlerului de robot KR C1, [74]

Fig.4.11. prezintă arhitectura întregului sistem robot. D.p.d.v. hardware, utilizează o placă de bază PC cu **microprocesor Pentium** (de minim P166) și memorie RAM de 32 MB. Cele două sisteme de operare sunt rulate pe același procesor, schimbul de date între ele (cum sunt valori de variabile, comenzi, încărcare/descărcare programe robot) se realizează prin utilizarea protocolului TCP/IP. Funcțiile de operare, afișare și gestiune date sunt implementate sub **Windows95** (este disponibilă o interfață utilizator completă). **VxWorks** este utilizat pentru toate task-urile de timp real cum sunt: generarea traiectoriei; procesarea comenzilor și procesarea informațiilor provenite de la interfețe periferice (senzori).

Pentru comanda acționărilor respectiv cuplarea magistralelor și intrărilor/ieșirilor (I/O) digitale, se utilizează o **placă multifuncțională MFC** (Multi-Function Card) care se introduce pe magistrala PCI a plăcii de bază. Placa MFC este prevăzută cu un procesor de semnale digitale DSP (Digital Signal Procesor) care transformă PC-ul într-un controler industrial. Această placă de asemenea conține un circuit de “accelerator de timp real” (Real-time Accelerator Chip).

Pentru comunicația cu echipamente/dispozitive periferice, în configurația standard, un modul de **magistrală CAN** (care suportă protocolul DeviceNet), este integrat în placa MFC. Cuplarea altor sisteme de magistrale de câmp (Interbus, Profibus, FIP, etc.) este posibilă prin utilizarea unor plăci de extensie adiționale. Pentru comunicație prin Ethernet (utilizând protocolul TCP/IP) există două soluții:

1. o legătură cu placa MFC, realizând astfel un acces direct la sistemul de operare VxWorks;
2. o legătură prin placa de bază, realizând astfel accesul la schimbul de date între Windows95 și VxWorks.

De asemenea este suportată **interfața serială tipică PC** (de exemplu pentru integrarea senzorilor). Însă datorită cerințelor de funcționare în timp real, pentru aceasta se utilizează un protocol special.

4.3. Dezvoltarea echipamentelor de comandă numerică (CNC) cu arhitectură deschisă

În cazul echipamentelor de comandă numerică (CNC), solicitarea din partea integratorilor de sistem, pentru dezvoltarea unor arhitecturi deschise, a apărut mult mai devreme decât la controlerele de roboți. În consecință, în timp au fost dezvoltate mai multe variante, care oferă soluții cu arhitecturi mai mult sau mai puțin deschise.

4.3.1. Clasificarea echipamentelor CNC cu arhitectură deschisă

Echipamentele CNC “deschise” fabricate recent, pot fi clasificate în 3 mari categorii, fig.4.12.:

1. echipamente CNC dotate **cu interfețe utilizator deschise**,
2. echipamente CNC **cu acces limitat la “kernel”**,
3. echipamente CNC **cu arhitectură deschisă de comandă**.

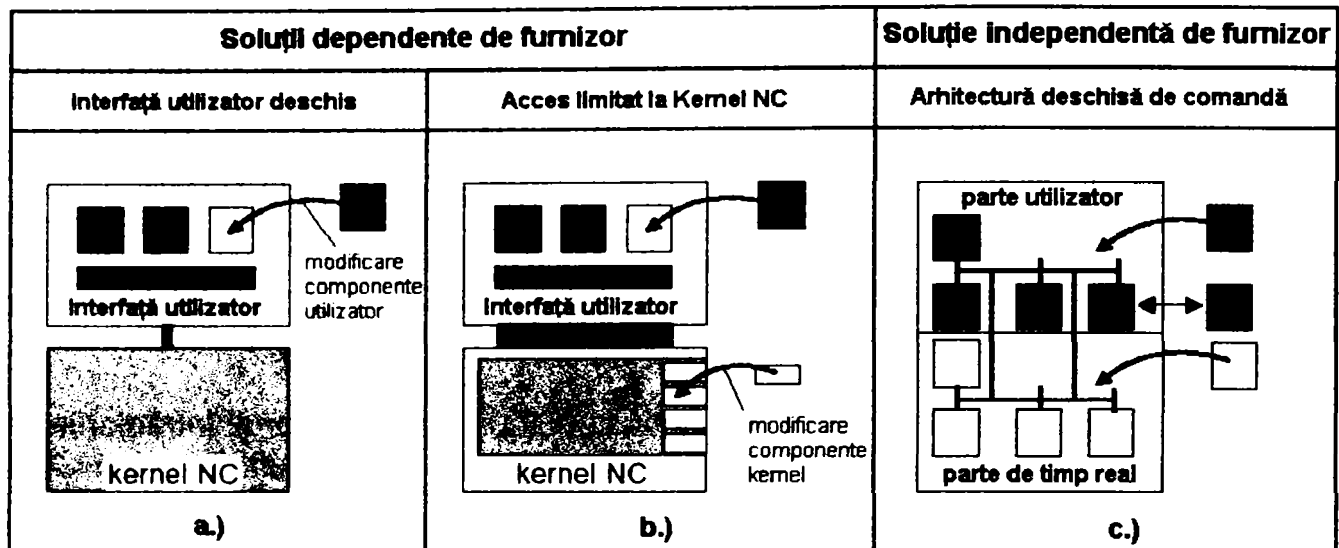


Fig.4.12. Clasificarea echipamentelor CNC "deschise", [33]

1. Trebuie arătat faptul că, deja aproape fiecare CNC nou este prevăzut cu o **interfață utilizator deschisă** (varianta a., din fig.4.12.), ceea ce a devenit o cerință minimală la vânzarea acestor echipamente. Această variantă de sistem, în anumite condiții, oferă posibilitatea modificării/adaptării libere a interfeței respective, prin modificarea componentelor/modulelor software aferente. Însă "nucleul" software-ului de comandă (numit "**Kernel NC**"), prin care se realizează funcțiile principale de control numeric (de exemplu interpolare, comanda mișcărilor la axe etc.) este "închis". Accesarea acestui nucleu este posibilă pentru un utilizator, doar prin intermediul unei interfețe speciale de programare, oferit de producător.

Conform firmei GE Fanuc (care este cel mai mare producător de CNC pe lume), **un CNC "deschis" este de fapt un CNC integrat într-un PC**, [40]. GE Fanuc oferă mai multe soluții unde, CNC-ul este practic un modul de extensie introdus într-un PC, pe care este rulat Windows NT, 2000 sau CE. Un exemplu este CNC-ul din seria 150i-M, fig.4.13. Utilizatorul sau fabricantul de mașină-unealtă, poate rula pe acest sistem și programele proprii, care pot accesa resursele CNC dar fără posibilitate de modificare/adaptare a acestora.

2. Pe lângă existența unei interfețe utilizator deschise, la unele CNC-uri (variante b.), există posibilitate și pentru "**completarea**" funcțiilor nucleului **Kernel** respectiv **schimbarea/inlocuirea limitată a unor algoritmi**. Acest fapt însă este foarte puțin sprijinit de marile firme producătoare de CNC-uri.

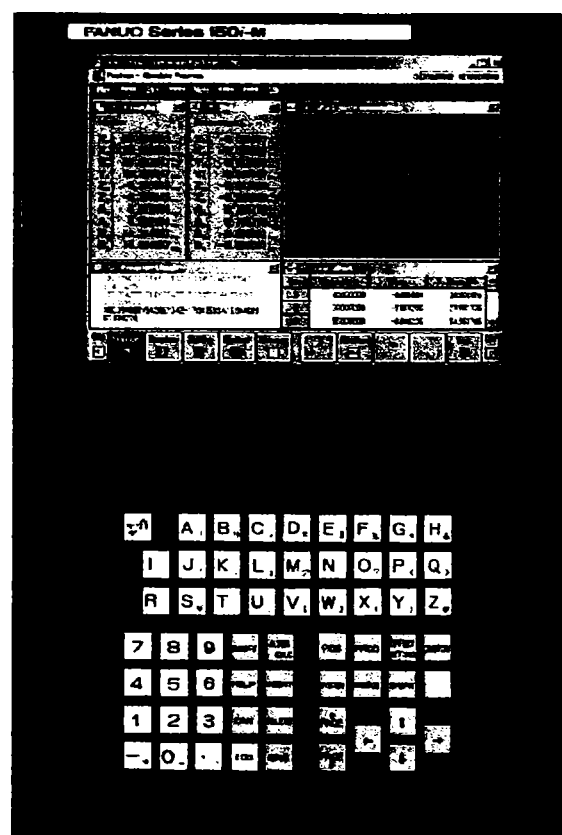


Fig.4.13. CNC Fanuc din seria 150i-M

Un exemplu este echipamentul **SINUMERIK 840C al firmei Siemens**, format din următoarele elemente funcționale: MMC (legătură om-mașină), NCK (Kernel NC) și PLC (automat programabil). Caracterul deschis se manifestă la nivelul MMC, prin posibilitatea rulării software-elor OEM pe o platformă WindowsNT, adică este aceeași soluție ca și în cazul firmei GE Fanuc. Suplimentar însă, cu licență OEM (adică cu contract direct cu producătorul), există posibilitatea integrării limitată a unor funcții speciale NCK, legate de interpolator, controlul de poziție etc.

Pe lângă "cei mari", din ce în ce mai multe firme mici oferă soluții complete pentru echipamente CNC pe bază de PC, cu arhitectură "parțial" deschisă. D.p.d.v. al arhitecturii utilizate există **3 configurații tipice**:

- Configurație în care, pe lângă placa de bază a PC-ului, nucleul Kernel este rulat pe un **modul de extensie**, cu conectare directă la acționări și module I/O. De exemplu familia XMP a firmei Motion Engineering, la care task-urile de timp real pot fi accesate prin intermediul unei interfețe C/C++, [81].
- Configurație cu două sau **mai multe procesoare** unde pe un procesor este rulat nucleul Kernel de timp real, iar pe altul este rulat interfața utilizator sub sistemul de operare Windows. Acest model este utilizat de Siemens și Fanuc.
- Configurație în care pe un **singur procesor** este rulat atât sistemul de operare Windows cât și Kernelul de timp real, în mod asemănător controlului de robot KUKA KR C1. Un exemplu în acest sens este sistemul OpenCNC® al firmei MDSI, unde pe un singur procesor este rulat WindowsNT și sistemul de operare RTX, [78]. Pe fig.4.14. se poate observa faptul că sistemul are atât interfață utilizator NT cât și de timp real.

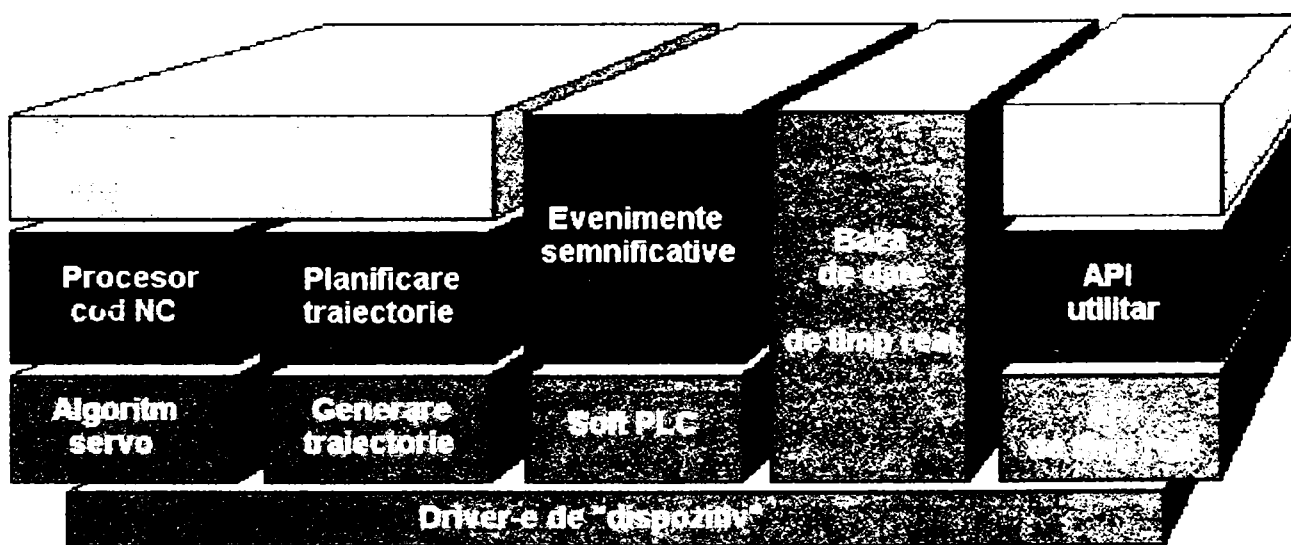


Fig.4.14. Arhitectura modulară a sistemului OpenCNC®

3. **Echipeamente CNC complet deschise** (varianta c.), sunt acele care au fost dezvoltate prin proiecte majore de cercetare internațională. În cadrul acestor proiecte de cercetare s-a urmărit obținerea **unor soluții independente de producător**, cu o arhitectură modulară, care să permită integrarea, modificarea, înlocuirea și interschimbabilitatea modulelor software atât în partea de utilizator, cât și în partea de timp real.

4.3.2. Proiecte majore de cercetare legate de dezvoltarea unor echipamente CNC cu arhitectură deschisă

Dezvoltarea unor echipamente CNC cu arhitectură deschisă a fost obiectul a mai multor proiecte de cercetare la nivel internațional. Cele mai importante sunt următoarele:

1. proiectul OSACA (Open System Architecture for Controls within Automation systems),
2. proiectul OMAC (Open, Modular Architecture Controller) și
3. proiectul OSEC (Open System Environment for Controllers).

4.3.2.1. Proiectul OSACA (Open System Architecture for Controls within Automation systems)

Proiectul OSACA (Open System Architecture for Controls within Automation systems) a fost lansat în anul 1992 cu finanțare ESPRIT. Până în anul 1998, cu un buget de 12,3 milioane de Euro, a dezvoltat prima oară în lume o arhitectură deschisă, independentă de furnizori.

În **consorțiul OSACA** au participat:

- mari întreprinderi producătoare de echipamente de comandă din Europa (Siemens, Bosch, NUM, Fagor, etc.,),
- întreprinderi producătoare de mașini-unelte (Homag, Comau, Trumpf),
- instituții de cercetare (ISW, WZL, IWB, INTEC etc.) și
- întreprinderi utilizatoare (BMW, DaimlerChrysler).

Prin proiectul OSACA s-a urmărit dezvoltarea unei arhitecturi modulare de control pentru mașini unelte. S-a pornit de la arhitectura de bază a sistemelor de control modular, care este formată dintr-o platformă de sistem (hardware-ul și software-ul de sistem) și dintr-un set de module ale software-ului de aplicație, care au implementate funcțiile specifice de control, fig.4.15.

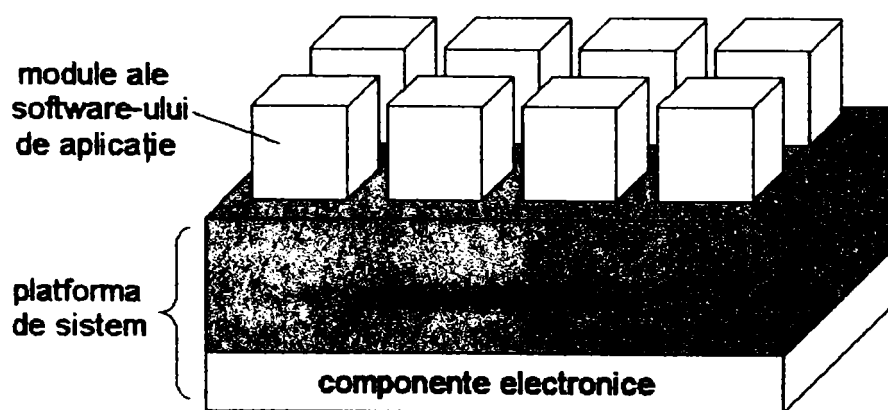


Fig.4.15. Structură de bază a unui sistem modular de control

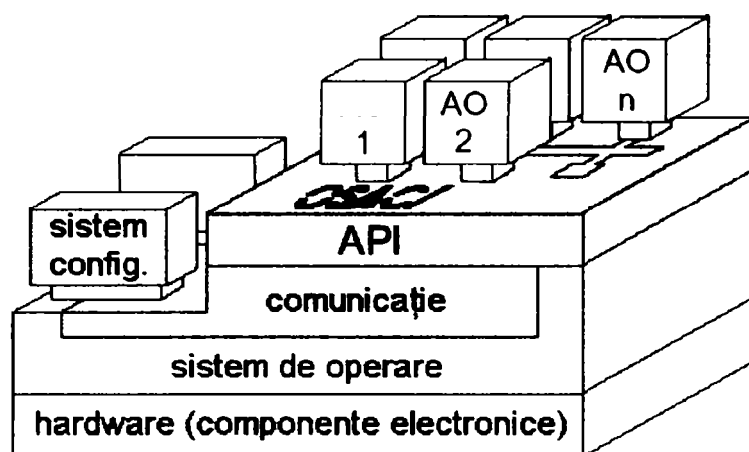


Fig.4.16. Principiul arhitecturii OSACA, [32]

Meritul cel mare al proiectului OSACA, este definirea unei interfețe de programare aplicație API (Application Programming Interface) care este independentă de furnizori. API asigură integrarea standardă a diferitelor module (AO – Application Object) și legătura lor la software-ul de sistem. Acesta din urmă este format din sistemul de operare, sistemul de comunicație și sistemul de configurare, fig.4.16.

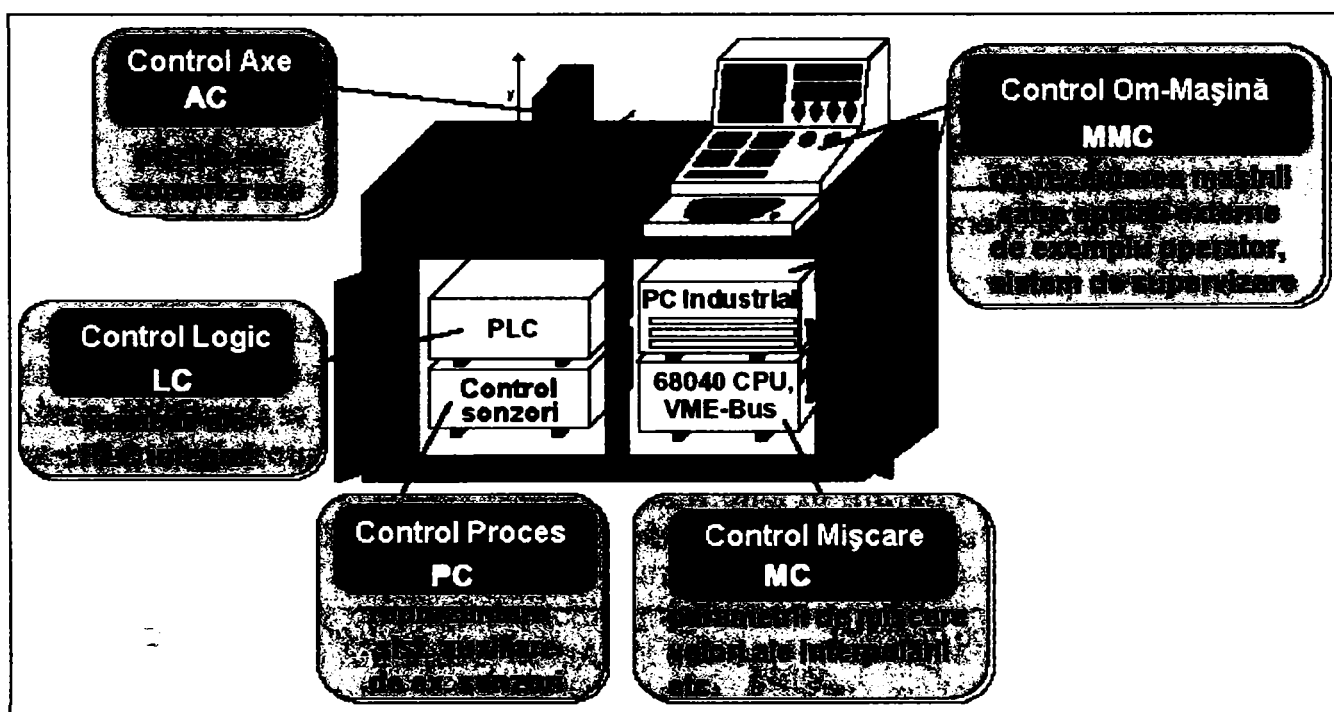


Fig.4.16. Elementele arhitecturii de referință OSACA, [32]

Arhitectura de referință OSACA, grupează obiectele de aplicație AO în cinci categorii:

1. **Control Om-Mașină (MMC – Man-Machine Control)** care reprezintă mașina sau o parte din mașină către entități externe, cum este un operator uman, sisteme de supervizare prin rețea LAN și sisteme CAD/CAM pentru mașină etc., respectiv permite acestor entități realizarea unui control prin operare.
2. **Control Mișcare (MC – Motion Control)** permite realizarea unor mișcări relative de-a lungul a mai multor grade de mobilitate, materializate prin axele mașinii. În urma controlului, mișcările rezultate trebuie să fie precise, continue și sincronizate.

3. **Controlul Logic** (LC – Logic Controls) este responsabil pentru funcționarea actuatorilor și pentru preluarea informațiilor de la senzorii montați pe mașină.
4. **Control Axe** (AC – Axis Control) include toate acele elemente care sunt necesare pentru activarea axelor în vederea execuției comenzilor de mișcare în condițiile prevăzute.
5. **Control Proces** (PC – Process Control) reprezintă - atunci când sunt utilizate - sistemele auxiliare ale mașinii. Este responsabil pentru gestiunea și procesarea datelor aferente acestor sisteme.

În general, modulele software corespunzătoare acestor grupuri de obiecte sunt rulate pe mai **multe platforme OSACA interconectate**. Un exemplu tipic este cazul unui echipament CNC cu arhitectură OSACA, la care panoul operator (având la bază un calculator compatibil PC) este conectat prin legătură Ethernet subTCP/IP la controlerul de mișcare (care are la bază magistrală VME), fig.4.17.

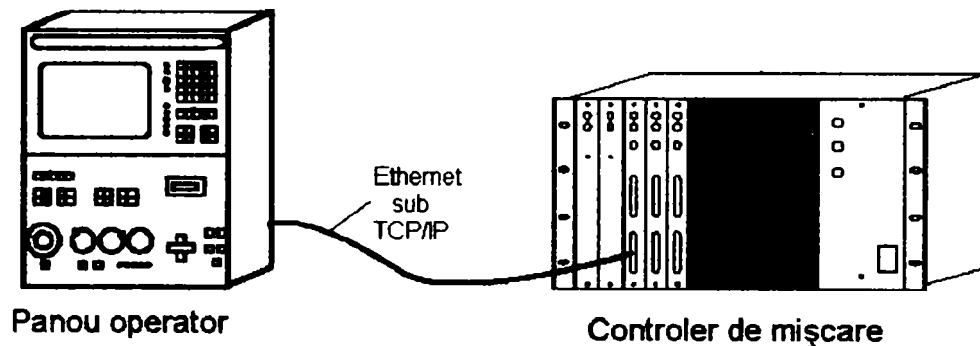


Fig.4.17. O configurație tipică OSACA

Sistemul deschis de control aferent, realizat pe baza arhitecturii OSACA, este constituit din două platforme: o platformă care are la bază un PC respectiv ca sistem de operare este utilizat Windows95/98, NT sau 2000 și o altă platformă cu procesor propriu, cu sistemul de operare în timp real VxWorks. Sistemul de comunicație permite modulelor AO, un schimb transparent de date prin legătura TCP/IP.

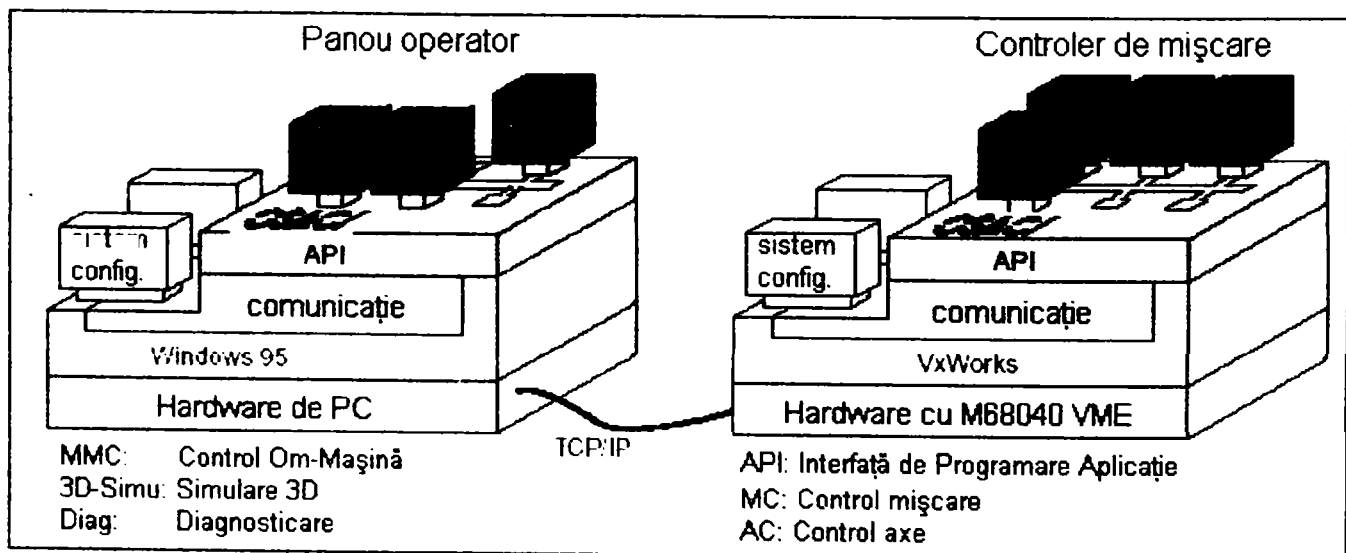


Fig.4.18. Arhitectura OSACA a sistemului

Existând deci o platformă API deschisă și bine definită, fiecare utilizator de echipament CNC (realizat pe baza arhitecturii OSACA), are acces la toate resursele acestuia, chiar și până la nivelul conducerii nemijlocite (de exemplu control și reglare poziție). Obiectele AO pot fi modificate, înlocuite sau pot fi integrate altele, suplimentare. Astfel, prin proiectul OSACA a rezultat o **arhitectură de comandă complet deschisă**, independentă de furnizori.

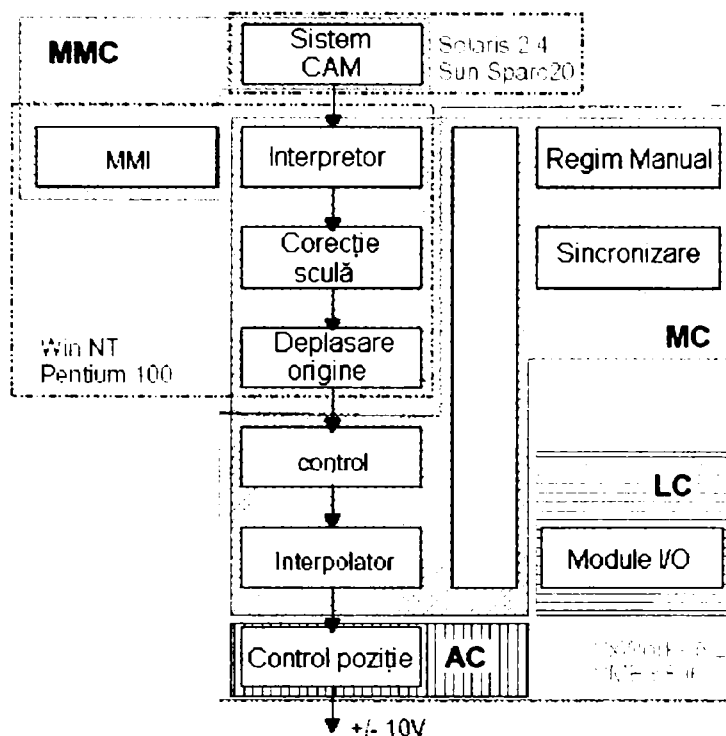


Fig.4.17. Structura de principiu a sistemului de comandă realizat la WZL

Primul sistem complet de comandă, cu structură OSACA, a fost realizat la institutul de cercetare WZL din Aachen. Structura de principiu a sistemului de comandă este prezentat în fig.4.17. Acesta utilizează 3 platforme OSACA interconectate: o platformă cu WindowsNT pentru operare, una cu VxWorks 5.2 pentru parte de control în timp real, și una cu Solaris 2.4. pentru partea de CAM.

Prototipuri ale sistemelor cu arhitectură OSACA au fost prezentate la mai multe universități și expoziții, chiar și în uzinele BMW și Mercedes. Firma Siemens a prevăzut că va lansa pe piață un echipament de comandă cu arhitectură OSACA. În-să datorită **lipsei de finanțare**, utilizatorii cei mai însemnați au rămas mediile universitare, unde aproape toate cercetările legate de CNC au loc prin utilizarea platformelor OSACA (de exemplu la WZL din Aachen, sau ISW din Stuttgart).

4.3.2.2. Proiectul OMAC (Open Modular Architecture Controller)

Proiectul OMAC (Open Modular Architecture Controller) a fost lansat în 1994, când trei întreprinderi constructoare de automobile din SUA, au publicat un sistem comun de cerințe legate de sistemele de comandă pe care le vor achiziționa în viitor, [21, 89]. Scopul a fost creșterea flexibilității în cadrul sistemelor proprii de fabricație, prin urgentarea și încurajarea furnizorilor în comercializarea controlerelor deschise respectiv modulare.

Trebuie arătat faptul că în cadrul proiectului nu s-a fixat o anumită arhitectură de referință ci a fost dezvoltată o metodologie și un sistem cadru general. În schimb s-a definit o serie de clase de bază care rezultă din decompoziția arhitecturii unui controler general. Clasele de bază sunt grupate în module, care sunt elementele de bază ale unui controler OMAC. Modulele sunt construite dintr-o serie de elemente predefinite respectiv sunt prevăzute cu niște interfețe de conectare. Modulele pot fi interconectate doar prin intermediul acestor interfețe.

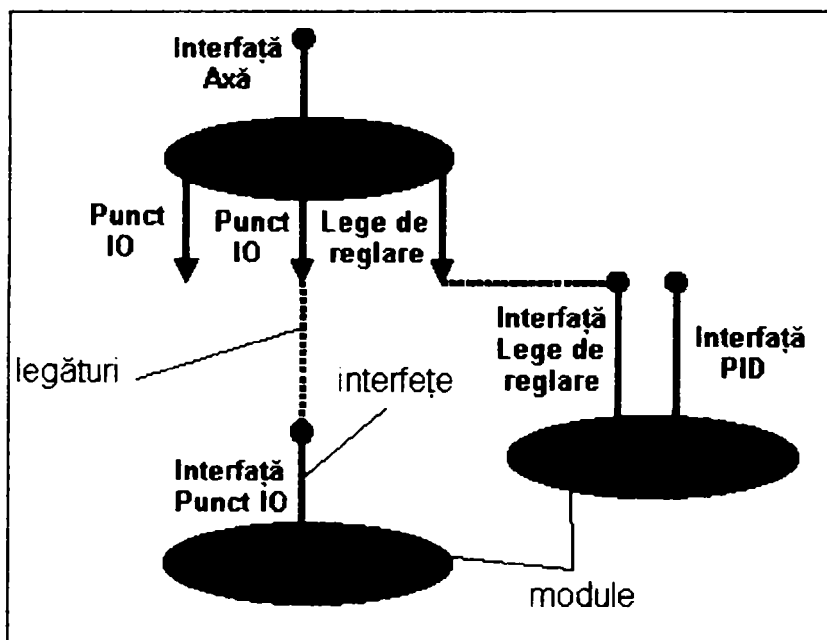


Fig.4.18. Exemplet de interconectare a modulelor prin interfețele aferente, [80]

În aceste condiții, construirea unui anumit sistem de comandă se reduce la selectarea, implementarea individuală și interconectarea diferitelor module OMAC. Având în vedere că au fost stabilite numai câteva reguli generale, proiectantul are mână liberă în stabilirea arhitecturii sistemului de comandă.

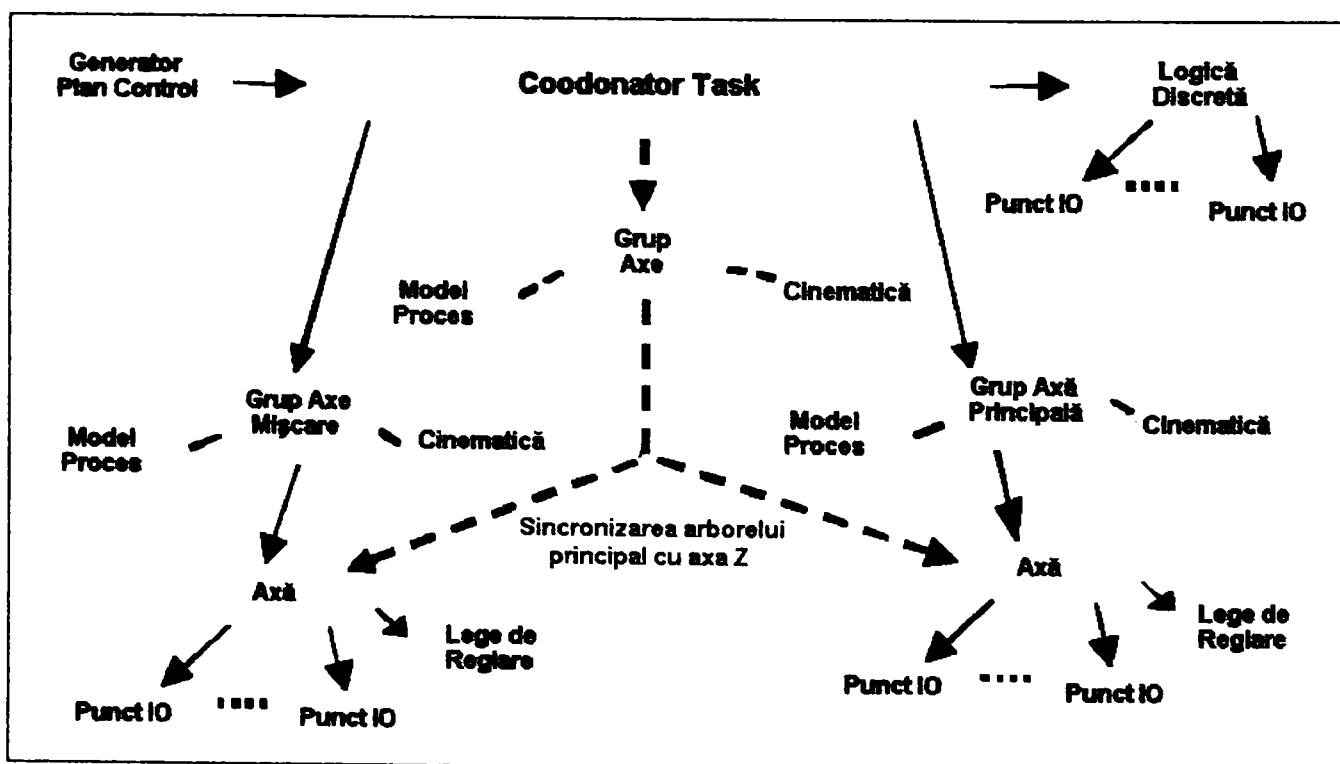


Fig.4.19. Modulele OMAC ale unei mașini de găurit, [80]

Din modulele OMAC poate fi construit orice echipament de comandă, de la cel mai simplu (utilizat de exemplu pentru comanda unui sistem de ridicare cu o singură axă), până la cel mai complex (utilizat de exemplu pentru comanda unui centru de prelucrare cu multe axe). În fig.4.19. s-au reprezentat modulele OMAC aferente sistemului de comandă al unei mașini de găurit, la care se poate sincroniza mișcarea axei Z cu mișcarea arborelui principal.

Asemănător proiectului OSACA, în prezent aplicațiile practice ale proiectului OMAC există doar la nivel de laboratoare de cercetare.

4.3.2.3. Proiectul OSEC (Open System Environment for Controllers)

În decembrie 1994 firmele Toyoda, Toshiba, Mazak și Mitsubishi au format un grup de lucru sub numele de OSEC (Open Sytem Environment for Controllers) cu scopul dezvoltării unei arhitecturi deschise de comandă specific japonez. Mai târziu, împreună cu mai multe firme și universități japoneze, care lucrau la dezvoltarea unor sisteme deschise de automatizare (numit FA - Factory Automation), au înființat grupul *JOP (Japan FA Open Sytem Promotion Group)*.

În cadrul acestui proiect a fost dezvoltat **modelul de referință OSEC**, care în mod asemănător nivelelor OSI este format din 7 nivele, [92]. Nivelele sunt următoare: mecanică, electrică, comandă dispozitiv, comandă geometrie, comunicație, operare precum și CAD/CAM. În centrul atenției au stat schimburile informaționale între aceste nivele.

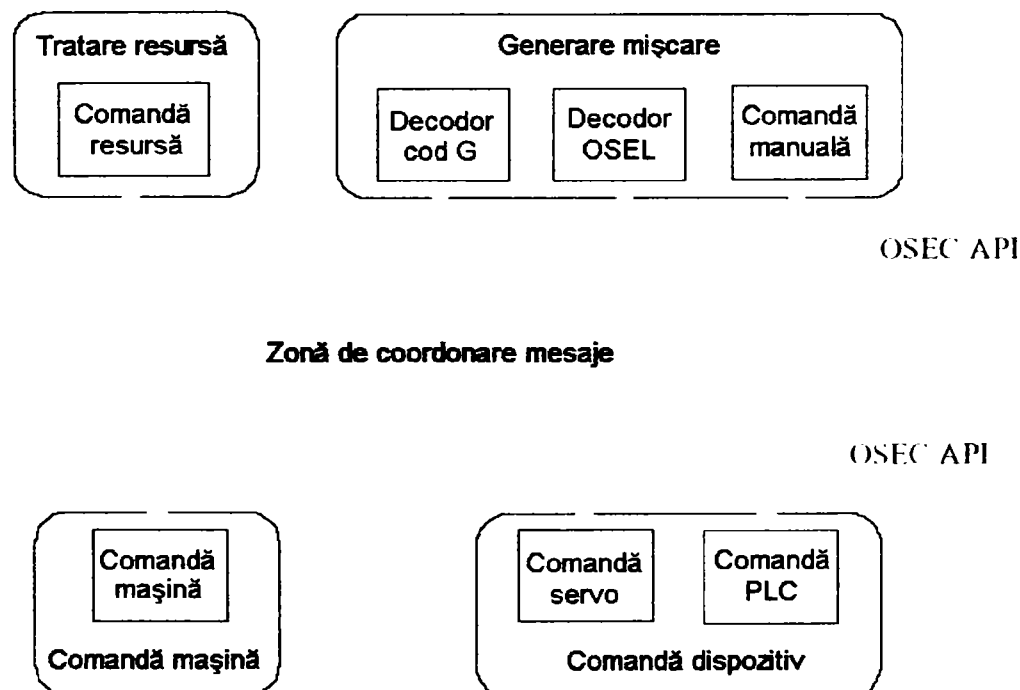


Fig.4.20.Arhitectura de referință OSEC, [84]

În mod asemănător proiectului OSACA, și în cazul arhitecturii OSEC resursele sistemului de comandă sunt grupate în module funcționale. În 1998 a fost elaborată o interfața de programare aplicație (API), care presupune existența între modulele fun-

cționale respective, a unei zone de coordonare mesaje, fig.4.20. În cadrul interfeței API sunt definite câteva obiecte generale, precum și modalitățile de acces la serviciile aferente acestor obiecte. Tot în cadrul acestui proiect a fost elaborată în 1999 o specificație cu numele PAPI care este interfața programului utilizator, adică specifică legătura între partea de comandă și partea de afișare/reprezentare a echipamentului.

Se poate considera că arhitectura OSEC-JOP este mult mai simplă decât arhitectura OSACA, însă la fel, aplicațiile practice există numai la nivel de laboratoare de cercetare.

4.4. Concluzii

În cadrul acestui capitol s-au analizat direcțiile de dezvoltare ale echipamentelor de comandă cu arhitectură deschisă. S-a subliniat faptul că dezvoltarea și introducerea în Hipersisteme CIM a echipamentelor cu arhitectură deschisă este o solicitare formulată de integratorii de sistem.

O arhitectură de comandă într-adevăr deschisă este caracterizată prin faptul că orice utilizator are acces la modulele software-ului de comandă, le poate modifica, adapta sau poate integra module noi în funcție de necesitățile procesului de fabricație. La baza realizării arhitecturii deschise de comandă, în majoritatea cazurilor se află tehnologia PC, atât la controlere de roboți cât și la echipamente de comandă numerică CNC. Aceasta tehnologie contribuie la obținerea unor echipamente de comandă cu preț scăzut.

Pentru asigurarea unei platforme deschise de operare se utilizează diferite variante ale sistemului de operare Windows. Pentru execuția task-urilor de timp real se utilizează un sistem de operare în timp real, de exemplu VxWorks.

În cazul controlerelor deschise de robot practic numai interfața utilizator este deschisă. Fabricantul în general furnizează o serie de biblioteci și module software, care pot fi utilizate de integratorii de sistem pentru crearea unor interfețe utilizator adecvate procesului de fabricație.

La echipamente CNC, pe lângă o interfață deschisă de utilizator, care este deja o cerință minimală la vânzarea acestora, în unele cazuri există posibilitatea realizării a unui acces limitat la nucleul software-ului de comandă. Aceasta însă este posibilă doar prin intermediul unor contracte OEM direct cu firma producătoare.

Pentru dezvoltarea unor echipamente cu arhitectură "complet" deschisă au fost lansate mai multe proiecte de cercetare la nivel mondial. Aceste proiecte vizează dezvoltarea unor sisteme de comandă CNC independente de producători. Deși existau o serie de demonstrații cu echipamente prototip, totuși aplicațiile practice au rămas numai la nivelul laboratoarelor de cercetare.

CAPITOLUL 5.

ELABORAREA UNOR MODELE DESTINATE STRUCTURILOR DESCHISE DE COMANDĂ A ROBOȚILOR INDUSTRIALI

5.1. Introducere

Având în vedere faptul că roboții industriali integrați în Hipersisteme CIM sunt în general de tipuri și de proveniență diferită, comanda de la distanță, adică **accesul de la distanță la controlul resurselor fizice și logice** ale fiecăruia în parte se soluționează în mod diferit. Motivul este că *fiecare producător în parte realizează o implementare particulară a funcțiilor de acces la aceste resurse*, [41]. Diferențe sunt chiar și la denumirea utilizată pentru “descrierea” resurselor. Astfel, la integrarea informațională a roboților industriali în Hipersisteme CIM în general apar costuri ridicate.

În aceste condiții a devenit necesară **standardizarea resurselor fizice și logice** ale roboților industriali (integrați în Hipersisteme CIM) respectiv **standardizarea modului de acces la aceste resurse**.

O încercare în acest sens au fost structurile MAP/MMS, prin intermediul standardului adițional ISO9506-3, care (din motivele prezentate în capitolul 3.) ulterior a fost retras. În cadrul acestui standard s-a încercat soluționarea problemei, prin definirea unui model virtual al robotului industrial care se află sub comanda de la distanță. Prin acest model, numit **Echipament Virtual de Fabricație Robot** (Robot Virtual Manufacturing Device - VMD), s-a stabilit un mod de acces standard la resursele echipamentului real, prin definirea unor obiecte și servicii MMS specifice roboților industriali.

Dezavantajul acestui model constă în faptul că pentru descrierea resurselor respectiv pentru accesul la resurse se utilizează structuri de date abstracte și (relativ) complicate. În aceste condiții, implementarea modelului VMD, în controlerul unui robot industrial, necesită un efort mare din partea producătorului, care de fapt este singurul în măsură să realizeze aceasta. În consecință majoritatea producătorilor de roboți industriali nu sprijină realizarea comenzii de la distanță prin intermediul structurilor MMS.

În acest capitol se propune o **nou mod de rezolvare a problemei**, bazat pe **concepția arhitecturii deschise**, care a fost prezentată în capitolul 4. Astfel, în mod asemănător echipamentelor CNC, se propune și pentru controlere de roboți industriali dezvoltarea unei arhitecturi deschise, independente de producător, caracterizată printr-o structură modulară a software-ului de sistem. Fiecare “modul” va controla una sau mai multe resurse ale robotului industrial. În acest mod accesul de la distanță la controlul resurselor se reduce la accesul standard la aceste module software.

Trebuie menționat că la stabilirea modulelor amintite respectiv a funcțiilor acestora s-a luat în considerare doar cerințele generale ale realizării comenzii de la

distanță a robotului industrial. Celelalte funcții, aferente numai comenzii locale, nu prezintă obiectul acestui capitol.

În concordanță cu cele de mai sus, s-a elaborat un **nou model de accesare de la distanță a resurselor fizice și logice ale unui robot industrial**, având la bază o arhitectură deschisă și modulară a software-ului de sistem.

Înainte de elaborarea acestui model, trebuiau stabilite acele resurse fizice și logice ale unui robot industrial (integrat într-un Hipersistem CIM) care pot fi sau în general sunt necesare să fie accesate prin comenzi de la distanță. Astfel au fost elaborate încă două modele:

- **Un model al resurselor fizice** ale roboților industriali integrați în Hipersisteme CIM. Acest model este o descriere generală a acelor resurse fizice la care se dorește a avea acces de la distanță.
- **Un model al resurselor logice** ale roboților industriali integrați în Hipersisteme CIM. Acest model, de asemenea este o descriere a acelor resurse logice la care se dorește a avea acces de la distanță.

5.2. Elaborarea unui model al resurselor fizice ale unui robot industrial

La elaborarea acestui model s-a urmărit stabilirea acelor resurse fizice, ale unui robot industrial integrat într-un Hipersistem CIM, care pot fi sau în general sunt necesare să fie accesate prin comenzi de la distanță.

Trebuie arătat că în cadrul standardului ISO9506-3 exista deja o încercare privind stabilirea acestor resurse fizice ale unui robot industrial. Ca **diferență** față de standardul amintit, modelul elaborat și prezentat în acest paragraf propune a nouă interpretare a resurselor fizice, respectiv presupune pentru controlerul de robot o "arhitectură deschisă de comandă", caracterizată printr-o **structură modulară a software-ului de sistem**. Astfel s-a încercat în plus stabilirea acelor "module", prin care se realizează controlul resurselor respective.

La elaborarea modelului, s-a plecat de la următoarele considerente:

1. Un robot industrial integrat într-un Hipersistem CIM, este "*un manipulator multiscop cu mai multe grade de mobilitate, care se află sub control automat, este reprogramabil și are o poziție fixă în cadrul halei industriale sau poate fi mobil*".
2. Funcția principală a unui robot industrial integrat într-un Hipersistem CIM, este **cea de manipulare**, de punere în mișcare materiale, piese etc. Aici trebuie menționat faptul că, chiar dacă robotul industrial realizează o operație tehnologică, de exemplu de vopsire sau de prelucrare, el efectuează în esență tot operații de "manipulare" (a pistolului de vopsit în cazul operației de vopsire sau a unei scule în cazul operației de prelucrare).
3. Robotul industrial integrat într-un Hipersistem CIM, poate fi considerat un **sistem automat** (industrial), care integrează mai multe subsisteme.

În concordanță cu cele de mai sus, s-au stabilit **subsistemele unui sistem robot**, (fig.5.3.). Astfel un "sistem robot" este compus din:

- **subsistem manipulator** (manipulator subsystem);
- **subsistem de conducere robot** (robot control subsystem);
- **echipament auxiliar** (auxiliary device);
- **program de aplicație robot** (robot application program).

Subsistemul manipulator este a cea parte a sistemului robot prin care se realizează operațiile fizice de manipulare. Subsistemul manipulator este pus în mișcare prin intermediul **subsistemului de conducere robot**. Acest subsistem, pe lângă conducerea (comandă și acționare) a mișcărilor subsistemului manipulator mai realizează și coordonarea acestora cu funcționarea unui **echipament auxiliar** (care este separat de subsistemul manipulator atât fizic cât și logic respectiv este necesar pentru realizarea unui anumit proces tehnologic de fabricație). Funcțiile de comandă și coordonare a întregului sistem robot sunt îndeplinite prin intermediul unui **program de aplicație robot**.

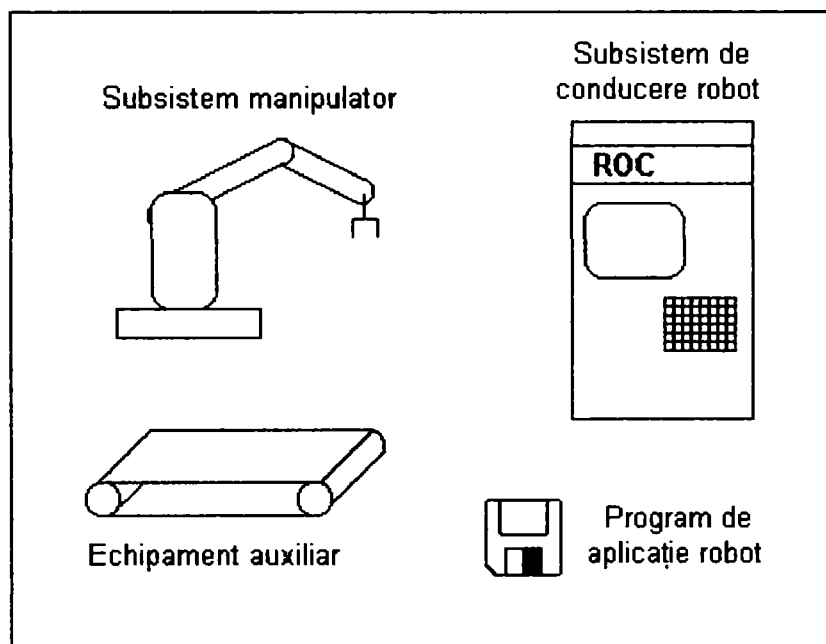


Fig.5.3. Sistemul robot și subsistemele acestuia

Într-un caz general un sistem robot poate să conțină **mai multe subsisteme de manipulare**, cu condiția ca acestea să fie sub coordonarea unui singur subsistem de conducere. În această accepțiune, un sistem robot va avea întotdeauna un singur subsistem de conducere.

De asemenea, trebuie menționat că într-un caz general un sistem robot poate avea în dotare **mai multe echipamente auxiliare**, care sunt coordonate de la același subsistem de conducere. Acesta din urmă poate să conțină, **mai multe programe de aplicație robot**, însă la un anumit moment dat, numai un singur "program" poate fi selectat pentru realizarea funcțiilor de comandă/coordonare a sistemului robot.

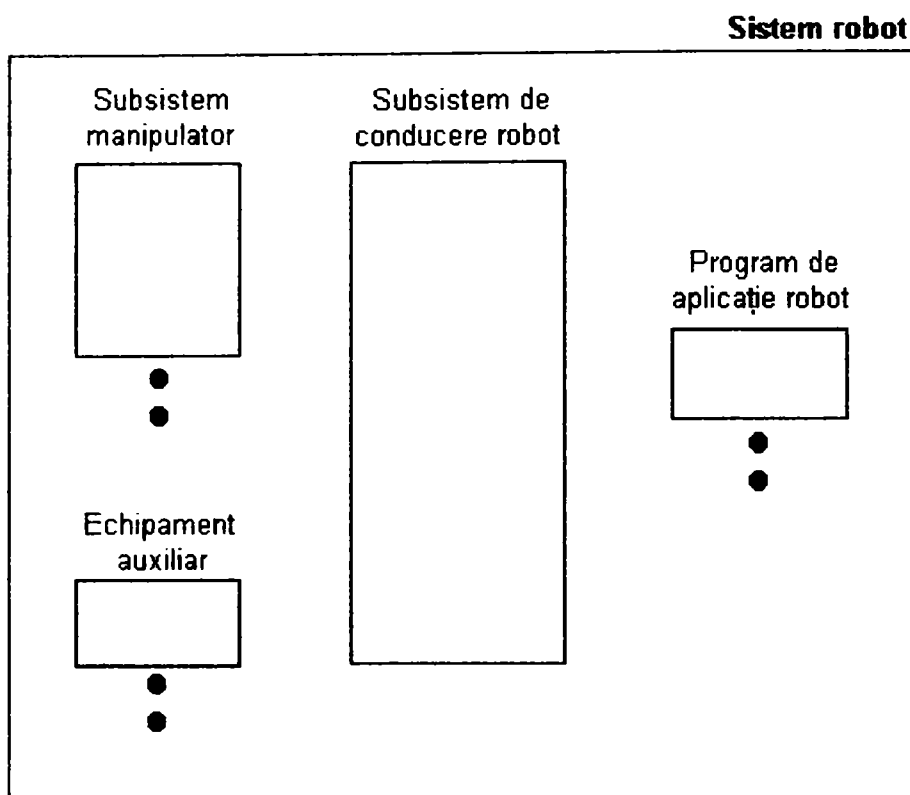


Fig.5.4. Modelul resurselor fizice al sistemului robot

Pe baza celor menționate, s-a întocmit fig.5.4, care reprezintă modelul resurselor fizice ale unui sistem robot.

5.2.1. Resursele fizice ale subsistemului manipulator

Elementul central al unui sistem robot este subsistemul manipulator. Conform standardului ISO9506-3, resursele fizice ale acestuia, care sunt necesare a fi accesate de la distanță sunt: **dispozitivul de ghidare** (robot arm) și **efectorul final** (end effector). În plus față de acesta, s-a considerat că în unele cazuri, subsistemul manipulator poate avea și **capabilități de deplasare** (locomotive capabilities), materializată prin intermediul unui **subsistem de deplasare** (locomotion subsystem). Un exemplu în acest sens pot fi roboții mobili. De asemenea s-a considerat că este necesară accesul de la distanță la această resursă fizică, ca urmare a fost inclusă în modelul elaborat, (fig.5.5.).

Dispozitivul de ghidare este acea parte a subsistemului manipulator care are funcția de a deplasa efectorul final, împreună cu obiectul care a fost solidarizat cu acesta, într-o anumită poziție, [70]. Este realizat sub forma unui lanț cinematic, care se compune dintr-un set de **elemente mecanice** (mechanical link). Elementele sunt racordate între ele constituind **cuple cinematice conducătoare** (joint) numite și articulații.

Fiecare cuplă cinematică conducătoare (prescurtat: c.c.c.) are numai un singur **grad de libertate**, adică admite o singură mișcare. Mișcarea permisă poate fi de rotație sau de translație. Astfel deosebim c.c.c. de translație (prismatic joint) respectiv de rotație (revolute joint).

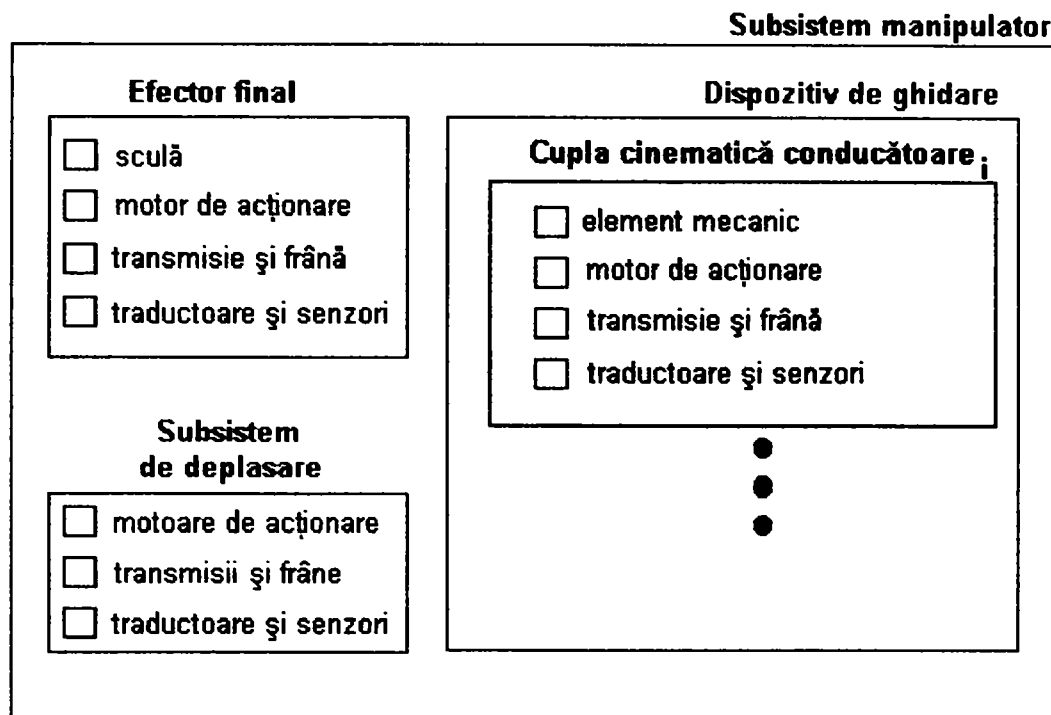


Fig.5.5. Modelul resurselor fizice ale subsistemului manipulator

Pentru punerea în mișcare a unui element mecanic, față de celălalt din componența c.c.c., se utilizează câte un **motor de acționare** (actuator) respectiv câte o **transmisie mecanică**, la nivelul fiecărei cuple cinematice conducătoare. Transmisia mecanică este utilizată pentru transmiterea mișcării de la motorul de acționare la elementul mecanic respectiv.

Se consideră că fiecare c.c.c. (numită și "axă a robotului") **poate fi sincronizată separat**, poate avea **limite de cursă superioară și inferioară** (upper and lower bounds), poate avea câte o **frână** (brake) și are asociat un set de **parametrii de legătură** (link parameters), prin care poate fi descrisă poziția relativă a elementului pus în mișcare.

Cuplele cinematice conducătoare sunt sub conducerea nemijlocită și independentă a unor **servosisteme de acționare** (servomechanism), specifice motoarelor de acționare, [77]. Există câte un servosistem de acționare separat pentru fiecare c.c.c. Un astfel de servosistem emite semnale de comandă către motorul de acționare și obține informații feedback de la **traductoarele** aflate la nivelul c.c.c.

Efactorul final are funcția de a solidariza obiectul manipulat de ultimul element al dispozitivului de ghidare, [70]. Poate fi format în general dintr-un dispozitiv de prehensiune, sculă sau cap de forță.

Observație:

- D.p.d.v. al comenzii de la distanță a sistemului robot, nu este necesară să se facă o distincție între cele 3 tipuri de efectoare finale. Astfel se propune utilizarea pentru acesta doar a termenului de **sculă** (tool).

Efactorul final poate fi sub acțiunea unui **motor de acționare** și poate conține traductoare și senzori utilizate în comanda motorului de acționare sau/și pentru pre-

luarea unor informații “feedback” de la mediu. La unele tipuri de roboți există posibilitatea înlocuirii automată sau manuală a efectorului final, cu un altul. Locul de “cuplare” al efectorului final, de ultimul element al dispozitivului de ghidare, este numit: **interfață mecanică** (mechanical interface).

Subsistemul de deplasare are funcția modificării situației sistemului robot în hala industrială în care este instalat. În general are în componență o serie de **motoare de acționare, transmisii, frâne, senzori și traductoare**.

Pe baza celor enunțate s-a întocmit fig.5.5. care reprezintă modelul resurselor fizice ale subsistemului manipulator.

5.2.2. Resursele fizice ale subsistemului de conducere

Un element esențial în cadrul unui sistem robot este **subsistemul de conducere** care se materializează în general sub forma unui **echipament de comandă robot** (ECR) numit și **controler robot** (robot controller - ROC). Acest subsistem are ca sarcină comanda întregului sistem robot prin:

- conducerea subsistemului manipulator;
- comanda echipamentului auxiliar;
- procesarea programului de aplicație;
- dialogul cu operatorul respectiv comunicația cu echipamente externe computerizate etc.

Pentru modelul resurselor fizice ale acestui subsistem, s-a propus o variantă diferită de cea stabilită în cadrul standardului ISO9506-3. Astfel s-a considerat pentru controlerul de robot o arhitectură modulară, unde fiecare “modul software” controlează una sau mai multe resurse ale robotului industrial.

Observații:

1. Așa cum s-a prezentat în capitolul 4., caracteristic unui controler “deschis” este o arhitectură modulară, care permite integrarea, modificarea, înlocuirea și interschimbabilitatea modulelor software atât în partea de utilizator, cât și în partea de timp real.
2. Din punctul de vedere al comenzii de la distanță esențial este că, prin accesarea acestor module se realizează un acces deschis și direct la controlul resurselor sistemului robot.

Astfel în concordanță cu cele prezentate, respectiv cu modelul subsistemului manipulator, s-a stabilit că subsistemul de conducere trebuie să conțină următoarele module software, fig.5.6.:

1. Controler dispozitiv de ghidare;
2. Controler efector final;
3. Controler echipament auxiliar;
4. Procesor program de aplicație;
5. Controler componente periferice;
6. Controler de comunicație;
7. Controler sistem robot.

Observații:

1. În cazul în care, subsistemul manipulator conține și un subsistem de deplasare, subsistemul de conducere robot mai conține și un modul de **controler de deplasare** (locomotion controller).
2. În cazul în care, sistemul robot conține mai multe subsisteme manipulator, atunci subsistemul de conducere robot integrează mai multe module de **controler dispozitiv de ghidare**, câte unul pentru fiecare.
3. În cazul în care, sistemul robot conține mai multe echipamente auxiliare, atunci subsistemul de conducere robot integrează mai multe module de **controler echipament auxiliar**, câte unul pentru fiecare.

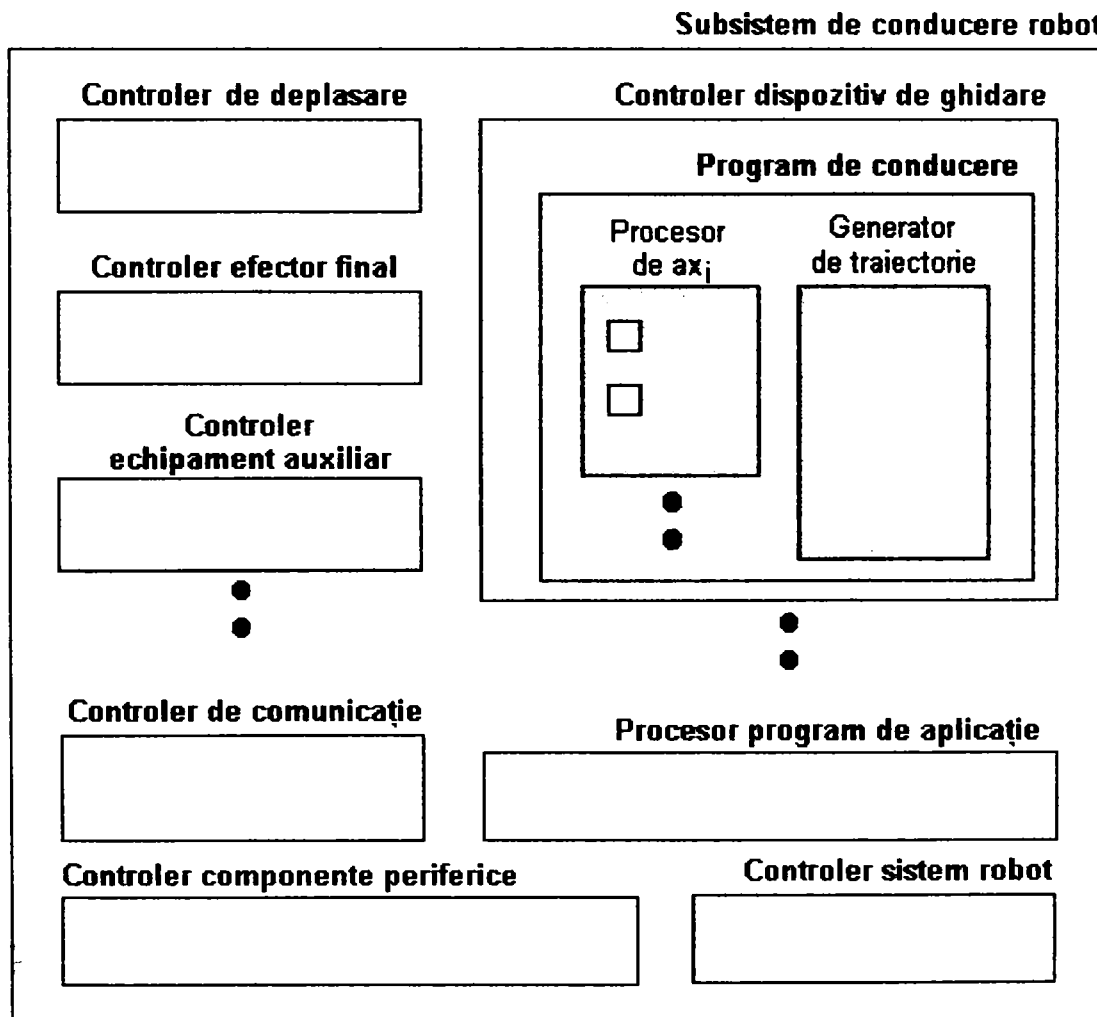


Fig.5.6. Modelul fizic al subsistemului de conducere robot

Controlerul dispozitivului de ghidare are ca funcție, conducerea mișcării dispozitivului de ghidare. Elementul esențial al acestui modul este **programul de conducere** (control program). Acesta de regulă are două părți componente:

1. un "subprogram" aferent **generatorului de traiectorie**;
2. și o serie de subprograme aferente **procesoarelor de ax**.

Generatorul de traiectorie acceptă comenzi de la **procesorul program de aplicație** (sau de la alte surse locale), pentru punerea în mișcare a dispozitivului de ghidare respectiv realizează convertirea acestor "cereri", într-un șir de comenzi peri-

odice (în timp), specifice procesoarelor de ax. În mod efectiv, generatorul de traiectorie este responsabil pentru convertirea unei traiectorii exprimate prin situarea, viteza și accelerația dorită a sculei (tool), în comenzi adecvate procesoarelor de ax. Pentru descrierea în mod constant a acestor mișcări sunt introduse o serie de sisteme de coordonate, care sunt descrise în paragraful 5.2.5.

Trebuie arătat că generatorul de traiectorie este responsabil și de controlul caracteristicilor dinamice ale mișcărilor dispozitivului de ghidare, [108]. Pentru realizarea comenzii de la distanță prezintă interes **controlul vitezei și accelerației vârfului de sculă** (adică a punctului caracteristic).

Observații:

1. **Viteza programată** (programmed speed) este viteza normală prescrisă generatorului de traiectorie, de către procesorul de program de aplicație. Aceasta este viteza dorită a vârfului de sculă și se stabilește în cadrul programului de aplicație.
2. **Factorul de viteză** (speed factor) este un factor de multiplicare, utilizat pentru modificarea uniformă (în sus sau în jos), a tuturor vitezelor programate, în cadrul programului de aplicație.
3. **Accelerația programată** (programmed acceleration) este accelerația normală prescrisă generatorului de traiectorie, de către procesorul de program de aplicație. Aceasta este accelerația dorită a vârfului de sculă și se stabilește în cadrul programului de aplicație.
4. **Factorul de accelerație** (acceleration factor) este un factor de multiplicare, utilizat pentru modificarea uniformă (în sus sau în jos), a tuturor accelerațiilor programate în cadrul programului de aplicație.

Valorile de intrare ale generatorului de traiectorie, includ situarea, viteza și accelerația programată a vârfului de sculă (exprimat în spațiul cartezian) și descriu **starea programată** a dispozitivului de ghidare.

Valorile de ieșire ale generatorului de traiectorie descriu **starea comandată** a dispozitivului de ghidare, exprimată în spațiul cuplelor cinematice conducătoare (joint space). Aceste valori de ieșire reprezintă, la un anumit moment dat, poziția, viteza și accelerația comandată a fiecărei cuple cinematice conducătoare în parte.

În fig.5.7. s-a încercat reprezentarea **fluxului de informații** existent la punerea în mișcare a dispozitivului de ghidare. Pe baza figurii, procesorul program de aplicație "prelucrează" programul de aplicație și pentru fiecare task al acestuia (care implică o modificare a situației curente), elaborează câte o **stare programată** a dispozitivului de ghidare, exprimată în valori "carteziene". Această stare programată este apoi transferată generatorului de traiectorie care "generează" prin calcul, seturi succesive de valori comandate.

Un set de valori (generat la un anumit moment dat), reprezintă o **stare comandată**, exprimată în spațiul cuplelor cinematice conducătoare. Stările comandate sunt transferate, la perioade egale de timp, procesoarelor de ax care prin comanda adecvată a motoarelor (de acționare), pun în mișcare relativă elementele c.c.c., cu vitezele respectiv accelerațiile comandate.

Prin monitorizarea **valorilor actuale** adică măsurate (ale pozițiilor, vitezelor etc.), se stabilește un control feedback al acestor mișcări la nivelul c.c.c.

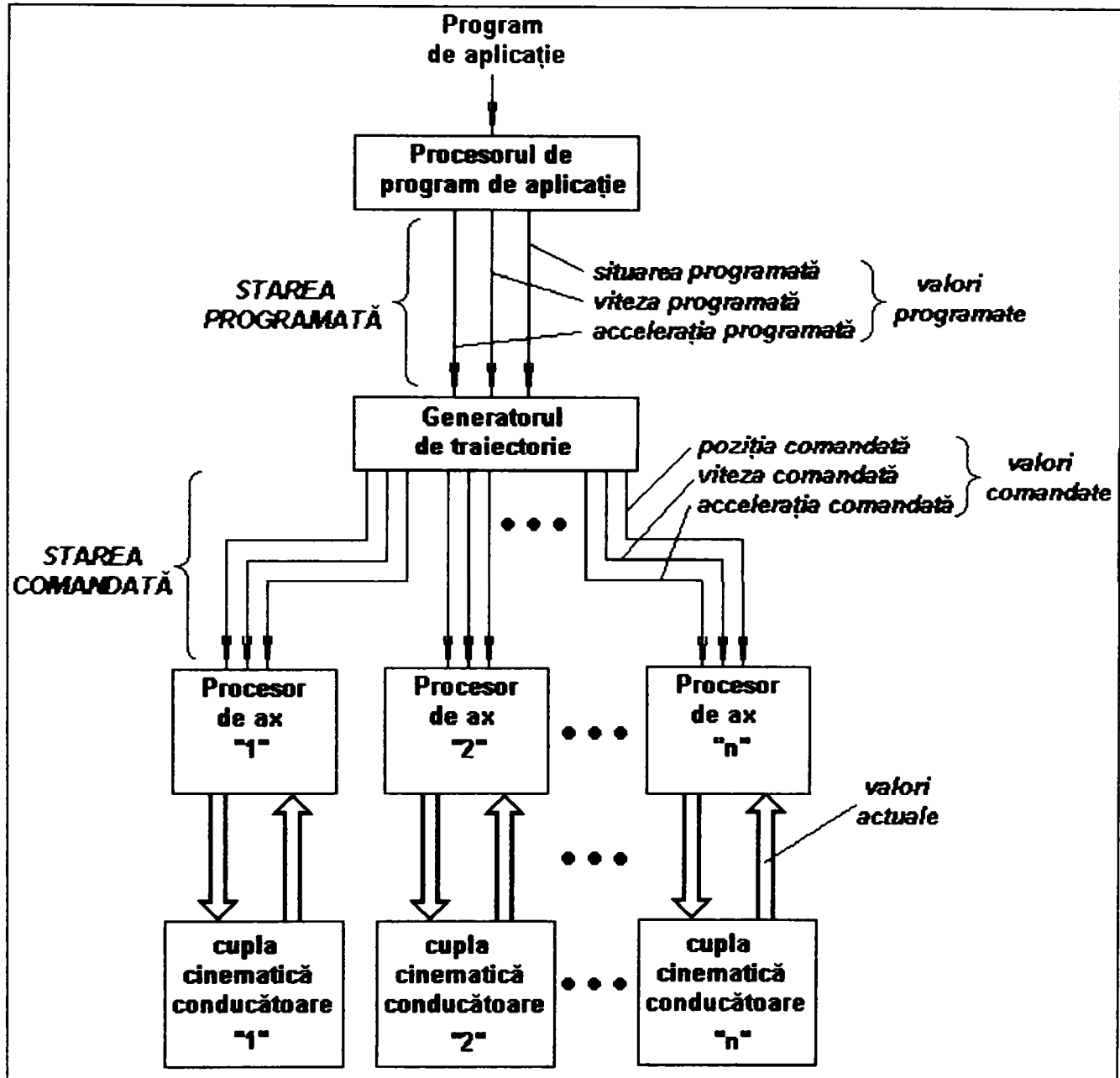


Fig.5.7. Fluxul de informații la punerea în mișcare a dispozitivului de ghidare

Controlerul efector final are ca sarcină comanda adecvată a motorului de acționare din componenta efectorului final, pe baza comenzilor furnizate de procesorul program de aplicație, respectiv pe baza informațiilor feedback, furnizate de traductoare și senzori.

Procesorul program de aplicație, în cadrul subsistemului de conducere robot, stabilește un mijloc pentru execuția programului de aplicație utilizator. Este de asemenea responsabil pentru generarea comenzilor de intrare pentru **controlerul de comunicație** (communication controller).

Controlerul componente periferice servește pentru comunicația subsistemului de conducere robot cu operatorul uman sau cu echipamente inteligente (com-

puterizate) externe. Ele sunt sub controlul controlerului de comunicație. Exemple de componente periferice:

- terminale,
- panou operator,
- panou de instruire,
- unități de stocare a informației (harddisk),
- module I/O de intrare și ieșire,
- interfață serială,
- interfață de LAN etc.

Contolerul sistem robot este un modul software prin care se realizează coordonarea funcționării tuturor subsistemelor din cadrul sistemului robot.

5.2.3. Echipamente auxiliare

Echipamentele auxiliare ale sistemului robot sunt subsisteme dependente de aplicație care sunt sub conducerea nemijlocită (directă) a subsistemului de conducere robot, prin intermediul unor module denumite **controler echipament auxiliar**. Exemple de echipamente auxiliare sunt:

- sisteme de comandă sudare,
- sisteme de vopsire,
- sisteme de control jet de apă,
- sisteme de vedere artificială, senzori etc.

Comanda echipamentelor auxiliare este integrată în programul de aplicație robot. Informațiile utilizate în comanda echipamentelor auxiliare, de exemplu parametrii specifici ai echipamentelor auxiliare, formează partea integrantă a parametrilor globali ai aplicației robot. Comanda echipamentelor auxiliare poate fi implementată într-un singur sau în mai multe programe din cadrul subsistemului de conducere robot. Echipamentele auxiliare sunt vizibile "de la distanță" ca o unitate separată, pot fi pornite sau oprite în mod independent de subsistemul manipulator.

5.2.4. Programul de aplicație robot

Prin intermediul **programului de aplicație robot** sunt prescrise de către un utilizator, într-un limbaj de programare specific subsistemului de conducere robot, task-urile care trebuie să fie efectuate de sistemul robot, în vederea derulării complete sau parțiale a unei anumite aplicații robotizate.

Într-un sistem robot poate fi încărcat, la un anumit moment dat, unu sau mai multe programe de aplicație robot. Odată încărcat, programul de aplicație robot **intră sub controlul procesorului program de aplicație** al subsistemului de conducere robot. **Selectarea** unui program de aplicație robot, în scopul lansării în execuție a acestuia, implică stabilirea unei conexiuni cu controlerul dispozitivului de ghidare.

Lansarea în execuție a programului de aplicație conduce la punerea în mișcare a subsistemului manipulator, de către modulul controler dispozitiv de ghidare.

5.2.5. Sisteme de coordonate robot utilizate la comanda de la distanță a subsistemului manipulator

Pentru realizarea unor structuri deschise de comandă de la distanță a roboților industriali integrați în Hipersisteme CIM este nevoie, pe lângă standardizarea resurselor fizice ale acestora și de stabilirea unui set standard de sisteme de coordonate, care sunt utilizate în descrierea mișcărilor subsistemului manipulator. Astfel s-a considerat că este necesară includerea în modelul resurselor fizice a următoarelor sisteme de coordonate standard (ISO9787), prezentate în fig.5.8., [62]:

- **Sistemul de Coordonate Universal (WORLD Coordinate System);**
- **Sistemul de Coordonate de Bază (BASE Coordinate System);**
- **Sistemul de Coordonate Interfață Mecanică (MICS - Mechanical Interface Coordinate System).**

Sistemul de Coordonate Universal este un sistem de referință fix, legat de hala industrială. Este utilizat în general pentru "layout"-ul celulei de fabricație. Definierea acestui sistem de coordonate este realizată în general de utilizatorul sistemului robot.

Sistemul de Coordonate de Bază stabilește locația plăcii de fixare a bazei robotului și este stabilit în general de furnizorul (fabricantul) robotului.

Sistemul de Coordonate Interfață Mecanică este un sistem de coordonate atașat interfeței mecanice, utilizat pentru cuplarea "sculei" (efectorului final), de ultimul element al dispozitivului de ghidare.

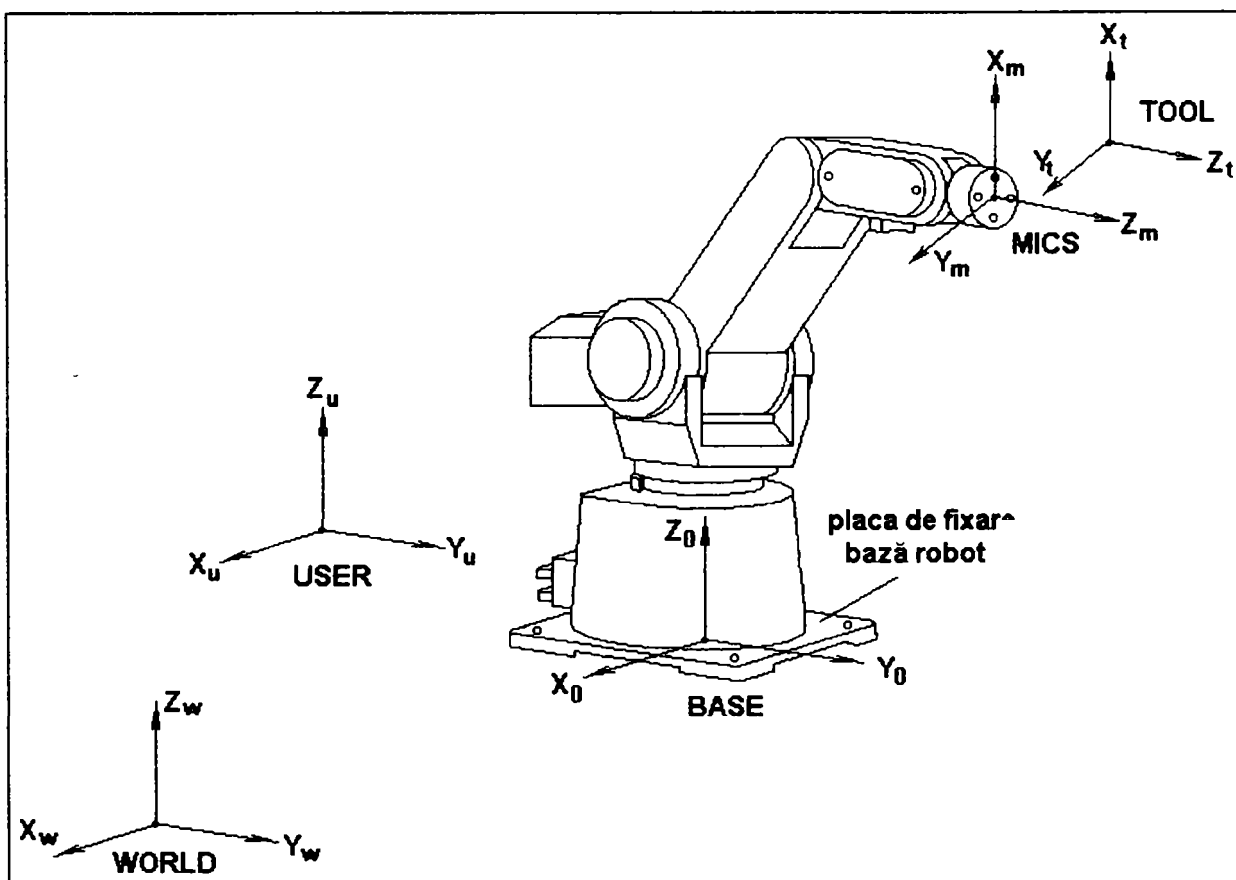


Fig.5.8. Sisteme de coordonate robot standardizate

Prin standardul ISO9506-3 au mai fost propuse încă 2 sisteme de coordonate robot care de asemenea au fost incluse în model:

- **Sistemul de Coordonate Sculă** (TOOL Coordinate System);
- **Sistemul de Coordonate Utilizator** (USER Coordinate System).

Sistemul de Coordonate Sculă este utilizat pentru definirea situației “sculei” (efectorului final). **Sistemul de Coordonate Utilizator** (User Coordinate System) este utilizat pentru obținerea unui sistem de referință alternativ pentru robot, altul decât sistemul de coordonate de bază.

5.3. Elaborarea unui model al resurselor logice ale unui robot industrial

La elaborarea acestui model s-a considerat că resursele logice sunt acele care nu au corespondent fizic (hardware sau software) bine determinat, în cadrul unui “sistem robot”. Ele **indică starea logică de funcționare a întregului robot sau numai a unor resurse fizice** așa cum poate fi observat de la celălalt capăt al “canalului de comunicație”.

Observație:

- Trebuie arătat că standardul ISO9506-3 nu a definit în mod expres vreun “model” al resurselor logice ale roboților industriali. S-a considerat că elaborarea acestuia este necesară totuși pentru dezvoltarea unui model de accesare de la distanță a resurselor (fizice și logice) ale roboților industriali integrați în Hipersisteme CIM.

S-a urmărit includerea în acest model a acelor resurse (stări) logice, a căror monitorizare de la distanță este necesară în cadrul unui Hipersistem CIM. Acestea sunt următoarele:

- capacitatea de funcționare;
- tipul controlului suportat;
- starea de sincronizare;
- starea de funcționare;
- starea de alimentare cu energie.

5.3.1. Capacitatea de funcționare

O resursă logică a cărei monitorizare este necesară prin comenzi de la distanță este cea referitoare la **capacitatea de funcționare** a sistemului robot. Această resursă se poate modela prin utilizarea a patru stări posibile. Cele patru stări sunt:

1. **OPERAȚIONAL**: nu este nici o defecțiune cunoscută.
2. **PARTIAL-OPERAȚIONAL**: nu poate fi realizată una sau mai multe funcții ale sistemului robot, datorită unor defecțiuni locale. De exemplu pot avea loc mișcările programate ale subsistemului manipulator însă este o defecțiune în funcționarea echipamentului auxiliar.

3. **NEFUNCȚIONAL**: sistemul robot nu mai poate funcționa datorită unor defecțiuni majore sau datorită lipsei de alimentare cu energie.
4. **NECESITĂ-Dispoziție**: Trebuie executată întâi o operație de dispoziție locală (o intervenție operator), ca sarcina curentă să poate fi îndeplinită.

Trebuie menționat faptul că stările **PARȚIAL-OPERAȚIONAL** și **NEFUNCȚIONAL** sunt rezultatul unor defecțiuni locale ale unor resurse fizice din cadrul sistemului robot.

5.3.2. Tipul controlului suportat

O altă stare logică foarte importantă pentru un robot industrial, integrat într-un Hipersistem CIM, este cea referitoare la **tipul controlului suportat** la un anumit moment dat. Astfel, în funcție de modul de realizare a controlului asupra funcționării robotului distingem două stări de funcționare:

- **cu comandă locală**,
- **cu comandă de la distanță**.

În cazul în care, la un anumit moment dat, sistemul robot permite **comandă locală**, atunci un operator uman prin intermediul unei console de programare, panou de instruire sau echivalent poate să emite comenzi locale pentru punerea în mișcare a robotului sau pentru selectarea și lansarea unui program de aplicație dorit. În cazul în care, la un anumit moment dat, sistemul robot nu permite acest mod de funcționare, robotul poate să ruleze un program de aplicație dar oprirea sau orice modificare a acestui program poate fi realizată doar prin **comandă de la distanță**.

5.3.3. Starea de sincronizare

Pentru ca sistemul robot să poate funcționa în regim automat, este necesar că după alimentarea cu energie (power-up), toate resursele fizice ale subsistemului manipulator să fie sincronizate.

Prin sincronizare vom înțelege o operație locală, prin care resursele fizice ale subsistemului manipulator sunt aduse pe rând în originile lor mecanice, în vederea obținerii unei coincidențe între poziția efectivă și poziția măsurată cu ajutorul transductoarelor. Pe baza celor de mai sus distingem 3 stări finite legate de starea de sincronizare:

- **NESINCRONIZAT**,
- **SINCRONIZARE-ÎN-CURS**,
- **SINCRONIZAT**.

Într-un caz general, operația de sincronizare poate fi inițiată atât prin comenzi locale (de exemplu prin acțiunea unui operator uman) cât și prin intermediul unor comenzi de la distanță.

5.3.4. Starea de funcționare

O altă problemă care trebuie rezolvată la comanda deschisă de la distanță a roboților industriali integrați în Hipersisteme CIM, este realizarea monitorizării stărilor de funcționare ale acestora. Pentru ca aceasta să fie posibilă trebuie întâi definite:

- stările automate finite, prin care poate fi modelată funcționarea unui sistem robot, precum și
- tranzițiile aferente trecerii dintr-o stare de funcționare în alta.

Stabilirea pentru un sistem robot a unor stări automate standard de a fost realizată prima oară în cadrul standardului ISO9506-3. Aceste stări, care au fost incluse și în modelul resurselor logice sunt următoarele:

- **ROBOT-INACTIV** (ROBOT-IDLE)
- **ROBOT-ÎNCĂRCAT** (ROBOT-LOADED)
- **ROBOT-READY**
- **ROBOT-ÎN-EXECUȚIE** (ROBOT-EXECUTING)
- **PAUZĂ-MIȘCARE-ROBOT** (ROBOT-MOTION-PAUSED)
- **NECESITĂ-INTERVENȚIE-MANUALĂ** (MANUAL-INTERVENTION-REQUIRED)

Starea **ROBOT-INACTIV** corespunde stării când în memoria echipamentului de comandă robot (ECR), nu este încărcat nici un program de aplicație. Această stare apare imediat după ce ECR este cuplat la sursa de energie sau când un program de aplicație a fost executat complet și programul respectiv este eliminat din memorie. Atunci când ECR este dotat cu baterie acumulator pentru memorarea programului de aplicație, această stare apare doar la eliminarea (ștergerea) programului de aplicație.

Starea de **ROBOT-ÎNCĂRCAT** corespunde unei stări când unu sau mai multe programe de aplicație sunt încărcate în memoria ECR.

Starea **ROBOT-READY** indică faptul că un program de aplicație disponibil a fost deja selectat, și ca urmare este "GATA" pentru execuție.

În starea **ROBOT-ÎN-EXECUȚIE** robotul este în "operație". Practic este rulat (este în curs de execuție) un program de aplicație care în unele faze poate genera comenzi pentru punerea în mișcare a subsistemului manipulator.

În starea **PAUZĂ-MIȘCARE-ROBOT** mișcarea robotului nu este permisă. Rularea programului de aplicație se poate continua sau se poate opri.

În starea **NECESITĂ-INTERVENȚIE-MANUALĂ** toate mișcărilor sunt oprite. Pentru execuția oricărei acțiuni, este necesară o comandă locală de către un operator uman. În urma acestei "intervenții" locale, respectiv "manuale", robotul intră în una dintre celelalte stări.

Trecerea sistemului robot dintr-o stare în alta se realizează prin intermediul unor tranziții de stare care pot fi atât rezultatul unor comenzi locale cât și a unor comenzi de la distanță.

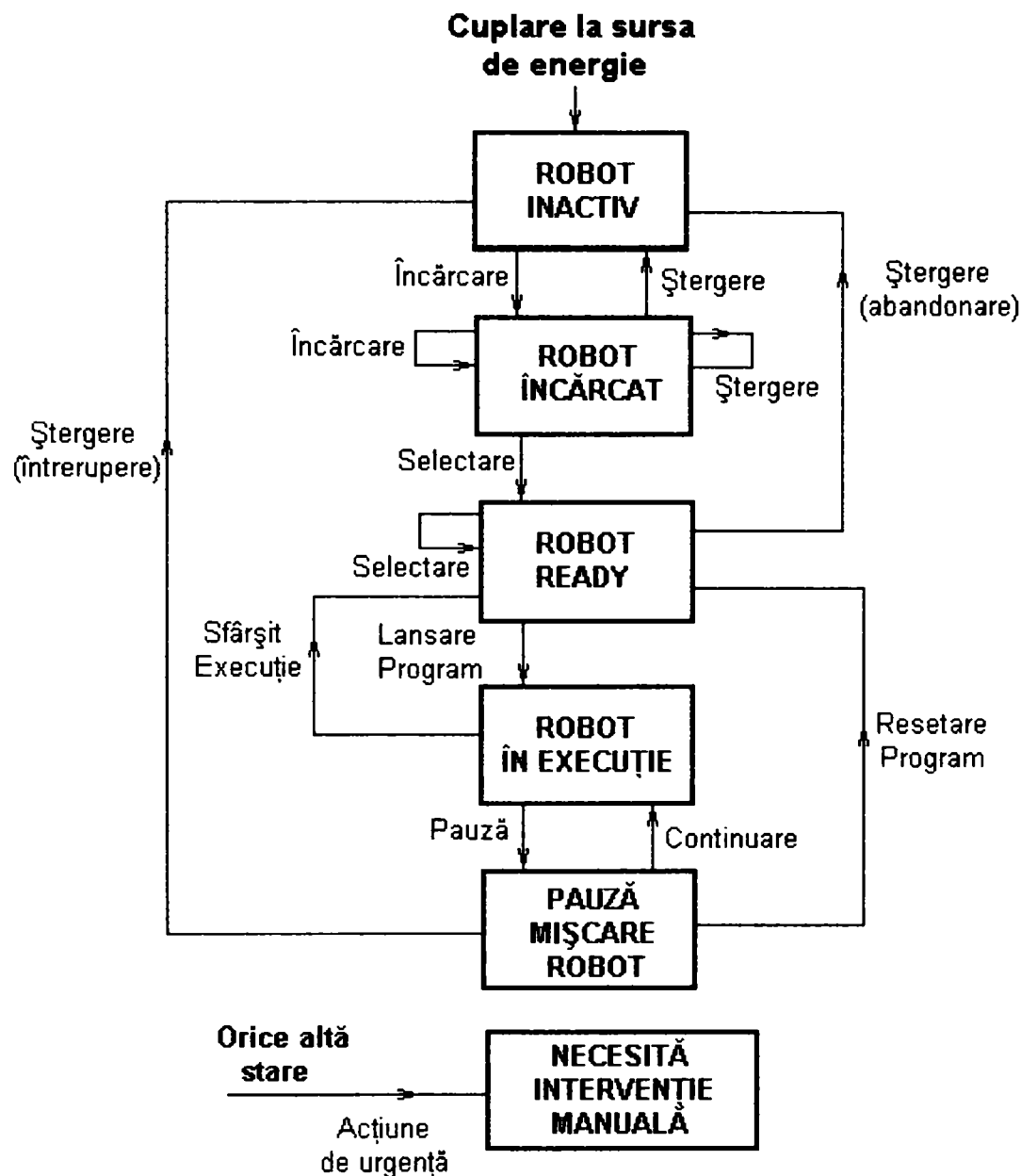


Fig.5.9. Diagrama stărilor de funcționare a sistemului robot

Diagrama stărilor de funcționare (care conține atât stările de funcționare cât și tranzițiile de stare aferente), adoptată pentru modelul resurselor logice, este reprezentată în fig.5.9. Trebuie menționat că aceasta **diferă de cea propusă în standardul ISO9506-3**. Diferențele sunt indicate în cursul descrierii tranzițiilor aferente:

- Tranziția **Încărcare** (Load) descrie acțiunea de încărcare a unui program de aplicație în memoria echipamentului de comandă robot ECR. Este de menționat faptul că în general ECR poate avea în memorie mai multe programe de aplicație, însă d.p.d.v. al acestei tranziții vom lua în considerare numai acele programe de aplicație care pot să conducă la punerea în mișcare a subsistemului manipulator. Dacă înainte de această tranziție, în memoria ECR, nu era nici un astfel de program de aplicație, atunci această tranziție cauzează (pe lângă încărcarea programului de aplicație) trecerea din starea ROBOT-INACTIV în starea ROBOT-

ÎNCĂRCAT. Dacă a fost cel puțin un astfel de program în memorie, atunci ca urmare a tranziției, robotul rămâne în starea **ROBOT-ÎNCĂRCAT**.

- Tranziția **Ștergere** (Clear) are loc atunci când, se realizează eliminarea (ștergerea) unui program de aplicație din memoria ECR. Dacă acest program a fost ultimul sau singurul program de aplicație în memoria ECR, atunci robotul intră în starea **ROBOT-INACTIV**. În caz contrar el rămâne în starea **ROBOT-ÎNCĂRCAT** cu programele de aplicație rămase în memorie. Tranziția Ștergere poate fi efectuată și din stările **PAUZĂ-MIȘCARE-ROBOT** respectiv **ROBOT-READY**. În acest caz toate programele de aplicație (chiar și cel care este selectat sau eventual este în curs de execuție), sunt eliminate din memoria ECR. În consecință robotul intră în starea **ROBOT-INACTIV**.
- Tranziția **Selectare** (Assign) se utilizează pentru pregătirea pentru execuție a unui anumit program de aplicație, din memoria ECR. Dacă inițial nu era nici un program "selectat", atunci această tranziție introduce robotul din starea **ROBOT-ÎNCĂRCAT** în starea **ROBOT-READY**. La ECR care pot conține în memorie numai un singur program de aplicație, tranzițiile **Încărcare** și **Selectare** au loc în cadrul unei singure operații.

Observație:

- Standardul ISO9506-3 a prevăzut și o tranziție de **Deselectare** (Deassign) prin care robotul trece din starea **ROBOT-READY** în starea **ROBOT-ÎNCĂRCAT**. S-a considerat că această tranziție nu este necesară în cazul în care selectarea unui alt program este posibilă și din starea **ROBOT-READY**, așa cum s-a prezentat în fig.5.9.
- Tranziția **Lansare Program** "de aplicație" (Program Start) este o operație fundamentală pentru punerea robotului în mișcare. Ca rezultat al acestei tranziții, robotul intră în starea **ROBOT-ÎN-EXECUȚIE**.
- Tranziția **Sfârșit Execuție** (End of Execution) apare ca rezultat al unor acțiuni locale din cadrul ECR. În urma acesteia robotul ajunge din starea de **ROBOT-ÎN-EXECUȚIE** în starea **ROBOT-READY**.

Observație:

- Standardul ISO9506-3 a prevăzut ca în urma tranziției **Sfârșit Execuție**, robotul să ajungă în starea **ROBOT-ÎNCĂRCAT**. În acest caz, înainte de lansare din nou în execuție, trebuia efectuată încă o tranziție suplimentară de **Selectare**. În varianta propusă aceasta nu mai este necesară, deoarece programul rămâne "selectat".
- Tranziția **Resetare Program** (Program Reset) servește pentru trecerea robotului aflat în starea **PAUZĂ-MIȘCARE-ROBOT**, în starea **ROBOT-READY**. Astfel sunt îndeplinite condițiile pentru lansarea programului respectiv de la început.
- Tranziția **Pauză** (Pause) cauzează întreruperea execuției programului de aplicație, dar menține toate valorile momentane ale contoarelor de ciclu și pași, la fel ca și starea logică din cadrul programului de aplicație.

- Tranziția **Continuare** (Continue) aduce robotul înapoi, din starea PAUZĂ-MIȘCARE-ROBOT în starea ROBOT-ÎN-EXECUȚIE.
- Tranziția **Acțiune de Urgență** (Emergency Action) poate să apară din orice stare și plasează robotul în starea NECESITĂ-INTERVENȚIE-MANUALĂ ca rezultat al unor acțiuni locale sau de la distanță. Un exemplu ar putea să fie apăsarea butonului de oprire de urgență (emergency stop). Revenirea din această stare nu poate să fie niciodată automată ci va necesita o acțiune explicită a unui operator uman de a plasa robotul într-o altă stare.

Trebuie arătat că funcția de **sincronizare a axelor** este considerată a fi executată într-o manieră similară unui program de aplicație normal, cu excepția că funcția de sincronizare nu plasează robotul în starea ROBOT-ÎNCĂRCAT.

5.3.5. Starea de alimentare cu energie

În cazul unui sistem robot, la punerea într-o anumită stare de funcționare, întâi este nevoie de alimentarea cu energie a subsistemului manipulator, a subsistemului de conducere respectiv după caz a echipamentelor auxiliare. Această alimentare se poate realiza simultan la toate subsistemele amintite sau succesiv. În acest ultim caz, întâi se alimentează cu energie electrică subsistemul de conducere, iar printr-o intervenție locală se alimentează cu energie și subsistemul manipulator. Astfel pentru fiecare subsistem în parte putem avea două stări:

- **Cuplat la sursa de energie** (Power ON),
- **Decuplat de la sursa de energie** (Power OFF)

Deși apare posibilitatea monitorizării separată a stării de alimentare cu energie a fiecărui subsistem în parte, s-a considerat că trebuie definită încă o stare, care se referă însă la întregul sistem robot:

- **Fiecare Resursă Fizică este Cuplată la sursa de energie** (All Physical Resource Power On),

Trebuie menționat că nu s-a definit separat o stare care să indice că "*fiecare resursă fizică este decuplată de la sursa de energie*" pentru că aceasta este identică cu starea **NEFUNCȚIONAL** descrisă în paragraful 5.3.1.

5.4. Elaborarea unui model de accesare de la distanță a resurselor fizice și logice ale unui robot industrial

Pornind de la cele două modele elaborate și prezentate în paragrafele anterioare, s-a conceput un **nou model de accesare de la distanță a resurselor fizice și logice ale unui robot industrial** integrat într-un Hipersistem CIM. Noutatea acestui model constă în faptul că propune rezolvarea problemei de acces de la distanță la resursele fizice și logice, pornind de la concepția arhitecturii deschise a controlerului robot.

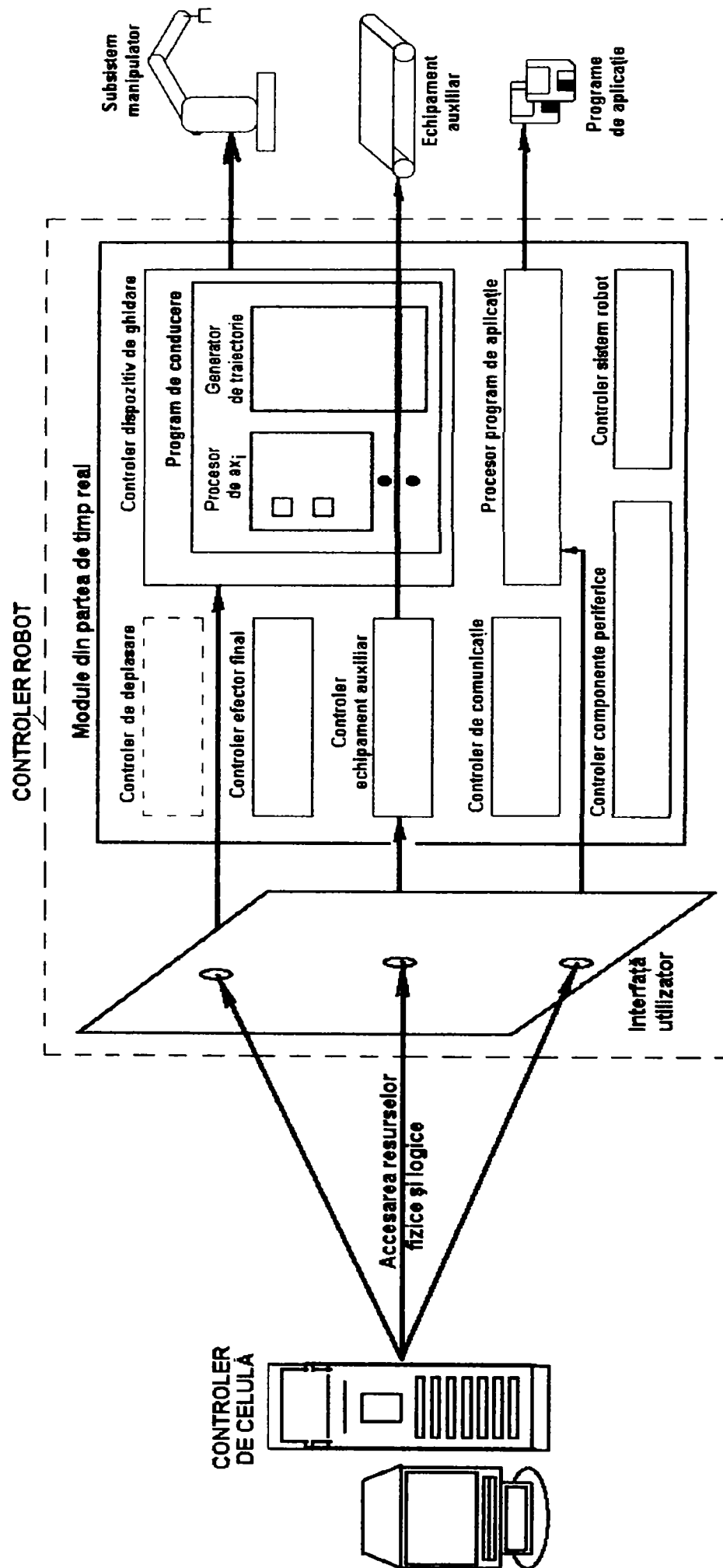


Fig.5.10. Principiul accesării de la distanță a resurselor fizice ale unui sistem robot conform modelului propus

5.4.1. Prezentarea modelului

În capitolul 4. s-a arătat că pentru echipamente CNC, există deja mai multe proiecte pentru dezvoltarea unor arhitecturi de comandă "complet" deschise. Modelul elaborat, urmărește această direcție a cercetărilor de pe plan mondial propunând și pentru controlere de roboți industriali dezvoltarea unei arhitecturi deschise, independente de producător, caracterizată printr-o structură modulară a software-ului de sistem. Aceasta trebuie să asigure accesul deschis la componentele/modulele software atât în partea utilizator cât și în partea de timp real a software-ului de sistem.

Trebuie menționat că la elaborarea acestui model s-a concentrat pe stabilirea unui acces standard la modulele software din partea de timp real (nu s-au analizat problemele specifice din partea utilizator). Principalele module din această parte, care sunt specifice conducerii sistemului robot, au fost stabilite în cadrul modelului resurselor fizice. Așa cum s-a arătat, prin fiecare modul se controlează una sau mai multe resurse ale robotului industrial. În acest mod accesul de la distanță la controlul resurselor se reduce la accesul standard la aceste module software. Astfel dacă se dorește accesul la controlul subsistemului manipulator, se accesează modulul "**controler dispozitiv de ghidare**", fig.5.10. Pentru "comanda" de la distanță a echipamentului auxiliar cu care este dotat sistemul robot, se accesează modulul "**controler echipament auxiliar**", etc.

Pentru accesul "de la distanță" la aceste module, s-a prevăzut o arhitectură client-server, bazat pe cereri de servicii (service request) emise de un echipament de comandă (de exemplu un controler de celulă aflat pe rol de Client) și pe servicii efectuate de controlerul de robot (aflat pe rol de Server), fig.5.11. Serviciile oferă practic un mijloc de acces la module software, care se finalizează prin accesul la controlul resursei aferente.

Trebuie arătat că stabilirea conținutului modulelor, prin acare se realizează controlul resurselor fizice ale sistemului robot, este o problemă legată de proiectarea și realizarea **sistemului de comandă locală**. Concentrându-se numai asupra problemei **comenzii de la distanță**, s-au stabilit funcțiile standarde ale principalelor module precum și "interfețele de acces" la aceste funcții.

Pentru accesarea de la distanță a funcțiilor standarde ale unui modul, s-a presupus că acesta, în secțiunea de interfață oferă o listă cu atributele modulului. Fiecare atribut reprezintă "poarta de acces" la o funcție standard de "control", implementat în cadrul modulului. De asemenea fiecare atribut poate fi accesat prin "citire" (READ), obținând astfel informații despre configurația sau starea curentă a resursei controlate. Unele atribute însă pot fi accesate și prin "scriere" (WRITE), astfel încât prin modificarea valorii atributului, practic se realizează un control asupra funcționării resursei respective.

Observație:

- Ideea reprezentării printr-o listă de "atribute", a interfeței de acces la funcțiile modulelor software, a fost preluată de la obiectele MMS introduse de standardul ISO9506 - 1 și 2. Însă funcțiile modulelor software respectiv mecanismele de acces la ele diferă de cele definite în standardul respectiv.

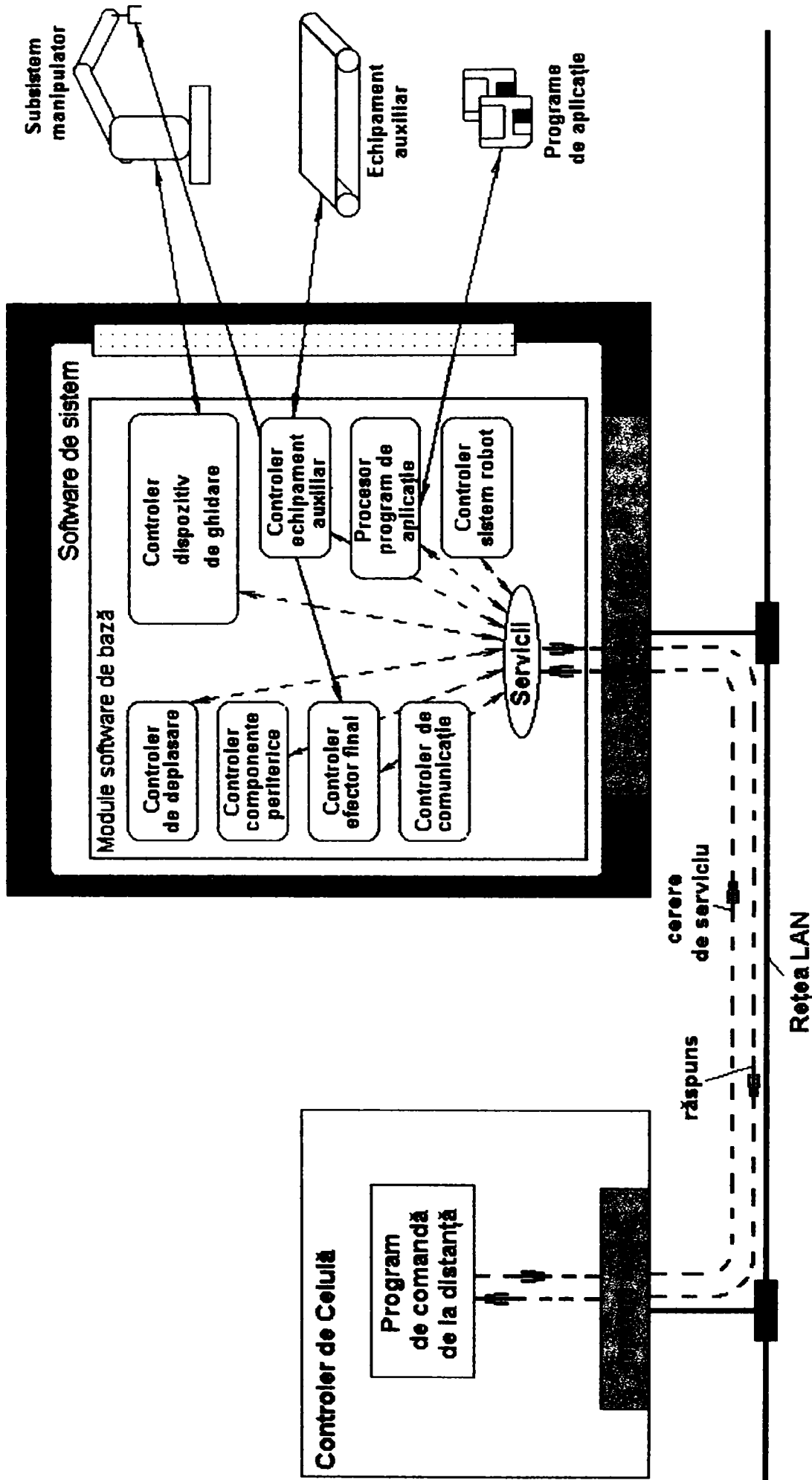


Fig.5.11. Modelul elaborat pentru accesarea de la distanță a resurselor fizice și logice ale unui sistem robot

Pe lângă accesarea atributelor, prin citire (READ) respectiv scriere (WRITE), s-a prevăzut o serie de servicii oferite de software-ul de sistem, legate de încărcarea (LOAD), ștergerea (CLEAR), selectarea (SELECT), lansarea în execuție (START), oprirea (STOP) și continuarea execuției (RESUME) a programelor de aplicație. De asemenea s-a prevăzut un serviciu "stare" (STATUS), care oferă informații despre starea curentă de funcționare a sistemului robot. Acest serviciu este identic cu accesarea prin citire a atributului Stare Funcționare Robot al modulului Controler sistem robot.

5.4.2. Stabilirea atributelor modulelor definite

La elaborarea modelului au fost stabilite "atributele" principalelor module ale software-ului de sistem, care sunt legate de controlul resurselor specifice domeniului roboților industriali. Ele sunt următoarele:

- Controler dispozitiv de ghidare;
- Controler echipament auxiliar;
- Controler echipament auxiliar;
- Procesor program de aplicație;
- Controler sistem robot.

Modulul *Controler dispozitiv de ghidare* conține o serie de atribute prin care se realizează un acces la *programul de conducere* al dispozitivului de ghidare; la subprogramul aferent *generatorului de traiectorie* respectiv la subprogramele aferente *procesoarelor de ax*.

MODUL:	CONTROLER DISPOZITIV DE GHIDARE
Atribut:	PROGRAM DE CONDUCERE
Atribut:	Tipul Controlului (LOCAL, DE-LA-DISTANȚĂ)
Atribut:	Cuplat la Sursa de Energie (ADEVARAT, FALS)
Atribut:	Starea de Sincronizare (SINCRONIZAT, NESINCRONIZAT, SINCRONIZARE-ÎN-CURS)
Atribut:	Număr Grade de Mobilitate
Atribut:	Situare Sistem de Coordonate Sculă (TOOL-MICS)
Atribut:	Situare Sistem de Coordonate Utilizator (USER-BASE)
Atribut:	Permis Mișcare (ADEVARAT, FALS)
Atribut:	Stare Mișcare (OPRIT- MIȘCARE, MIȘCARE-ÎN-CURS)
Atribut:	PROCESOARE DE AX
Atribut:	Listă de Descriere Procesoare de Ax
Atribut:	Numărul Procesorului de Ax
Atribut:	Tipul Cuplei Cinematice Conducătoare (ROTAȚIE, TRANSLAȚIE)
Atribut:	Echipat cu Frână (ADEVARAT, FALS)
Dacă:	Echipat cu Frână = ADEVARAT atunci:
Atribut:	Stare Frână (CUPLAT, DECUPLAT)
Atribut:	Starea de Sincronizare (SINCRONIZAT, NESINCRONIZAT, SINCRONIZARE-ÎN-CURS)
Atribut:	Limită Superioară Poziție
Atribut:	Limită Inferioară Poziție

<p>Atribut: Starea Comandată Cupla Cinematică</p> <p>Atribut: Poziția Comandată</p> <p>Atribut: Viteza Comandată</p> <p>Atribut: Accelerația Comandată</p> <p>Atribut: Starea Actuală Cupla Cinematică</p> <p>Atribut: Poziția Actuală</p> <p>Atribut: Viteza Actuală</p> <p>Atribut: Accelerația Actuală</p> <p>Atribut: GENERATOR DE TRAIECTORIE</p> <p>Atribut: Starea Programată</p> <p>Atribut: Situația Programată Sculă (TOOL-USER)</p> <p>Atribut: Factor de Viteză</p> <p>Atribut: Viteza Programată</p> <p>Atribut: Factor de Accelerație</p> <p>Atribut: Accelerația Programată</p> <p>Atribut: Situația Curentă Sculă (TOOL-USER)</p>

Fig.5.12. Interfața modulului Controler dispozitiv de ghidare

Structura interfeței modulului *Controler dispozitiv de ghidare* este prezentată în fig.5.12. Atributele acestui modul au următoarele semnificații:

- Atributul **PROGRAM DE CONDUCERE** conține o serie de “subatribute” prin care sunt descrise stările generale ale dispozitivului de ghidare:
 - Atributul **Tipul Controlului** indică faptul că dispozitivul de ghidare, este sub *comandă locală* sau este permisă *comanda de la distanță*.
 - Atributul **Cuplat la Sursa de Energie** este de tip boolean. Prin valoarea ADEVĂRAT sau FALS indică faptul că dispozitivul de ghidare este cuplat la sursa de energie sau nu.
 - Atributul **Starea de Sincronizare** indică faptul că dispozitivul de ghidare, ca o unitate, este sincronizat sau nu. Acest atribut are valoarea SINCRO-NIZAT dacă toate axele sunt sincronizate; are valoarea SINCRO-NIZARE-ÎN-CURS dacă axele sunt în curs de sincronizare iar în caz contrar are valoarea NESINCRO-NIZAT.
 - Atributul **Număr Grade de Mobilitate** este un număr de tip întreg și indică numărul cuplelor cinematice conducătoare din componența dispozitivului de ghidare.
 - Atributul **Situația Sistem de Coordonate Sculă (TOOL-MICS)** este de tip situație și indică prin valorile sale situația Sistemului de Coordonate Sculă (TOOL) față de Sistemul de Coordonate Interfață Mecanică (MICS).
 - Atributul **Situația Sistem de Coordonate Utilizator (USER-BASE)** este de tip situație și indică prin valorile sale situația Sistemului de Coordonate Utilizator (USER) față de Sistemul de Coordonate de Bază (BASE).
 - Atributul **Permis Mișcare** (Motion Enabled) este de tip boolean. Are valoarea ADEVĂRAT dacă dispozitivul de ghidare sau orice parte a acestuia poate fi pus în mișcare dacă apare un semnal valid de comandă din partea generatorului de traiectorie. Dispozitivul de ghidare se va mișca în mod autonom dacă și numai dacă acest atribut are valoarea ADEVĂRAT.

- Atributul **Stare Mișcare** indică prin valorile sale faptul că dispozitivul de ghidare sau orice parte a acestuia este pus în mișcare (MIȘCARE-ÎN-CURS) sau este oprit (OPRIT-MIȘCARE).
- Atributul **PROCESOARE DE AX** conține următoarele subatribute:
 - Atributul **Listă Descriere Procesoare de Ax** este o listă utilizată pentru descrierea pe rând a stărilor generale aferente fiecărui procesor de ax. Conține următoarele atribute:
 - Atributul **Numărul Procesorului de Ax** indică numărul de ordine al procesorului de ax respectiv. Acest număr este identic cu numărul de ordine al cuplei cinematice conducătoare aferentă procesorului de ax respectiv.
 - Atributul **Tipul Cuplei Cinematice Conducătoare** (Joint Type) indică tipul cuplei cinematice conducătoare aferentă procesorului de ax respectiv. În funcție de tipul cuplei cinematice conducătoare (de tip rotație sau de tip translație), atributul poate lua valoarea: ROTAȚIE sau TRANSLAȚIE.
 - Atributul **Echipat cu Frână** este de tip boolean și prin valoare indică faptul că există (ADEVĂRAT) sau nu există (FALS) frână la nivelul cuplei cinematice conducătoare. Dacă există frână atunci există și atributul **Stare Frână**. Acest atribut indică faptul că se utilizează (CUPLAT) sau nu (DECUPLAT) frânele pentru restricționarea mișcării.
 - Atributul **Starea de Sincronizare** indică starea cuplei cinematice conducătoare. Stările posibile sunt: SINCRONIZAT, NESINCRONIZAT sau SINCRONIZARE-ÎN-CURS.
 - Atributul **Limită Superioară Poziție** este o variabilă care indică prin valoare o coordonată corespunzătoare limitei superioare de mișcare pentru cupla cinematică conducătoare.
 - Atributul **Limită Inferioară Poziție** este o variabilă care indică prin valoare o coordonată corespunzătoare limitei inferioare de mișcare pentru cupla cinematică conducătoare.
 - Atributul **Starea Comandată Cuplei Cinematice** indică printr-un set de 3 valori **poziția**, **viteza** și **acclerația** comandată a cuplei cinematice conducătoare.
 - Atributul **Starea Actuală Cuplei Cinematice** indică **poziția**, **viteza** și **acclerația** actuală a cuplei cinematice conducătoare.
- Atributul **GENERATOR DE TRAIECTORIE** conține următoarele atribute:
 - Atributul **Starea Programată** indică printr-un set de valori starea programată a dispozitivului de ghidare:
 - Atributul **Situarea Programată Sculă (TOOL-USER)** indică prin valorile sale situarea țintă a Sistemului de Coordonate Sculă (TOOL) față de sistemul de Coordonate Utilizator (USER).
 - Atributul **Factor de Viteză** (Speed Factor) indică valoarea factorului de corecție a Vitezei Programate (Programmed Speed).

- Atributul **Viteza Programată** (Programmed Speed) indică valoarea vitezei programate a sistemului de coordonate al interfeței mecanice (MICS) față de sistemul de coordonate de bază, în cadrul programului de aplicație robot.
- Atributul **Factor de Acclerație** (Acceleration Factor) indică valoarea factorului de corecție a Acclerației Programate (Programmed Acceleration).
- Atributul **Acclerația Programată** (Programmed Acceleration) indică valoarea acclerației programate a sistemului de coordonate al interfeței mecanice (MICS) față de sistemul de coordonate de bază, în cadrul programului de aplicație robot.
- Atributul **Situarea Curentă Sculă (TOOL-USER)** este de tip situare și indică prin valorile sale situarea curentă a Sistemului de Coordonate Sculă (TOOL) față de Sistemul de Coordonate Utilizator (USER).

MODUL:	CONTROLLER EFECTOR FINAL
Atribut:	Număr Identificare
Atribut:	Cuplat la Sursa de Energie (ADEVARAT, FALS)
Atribut:	Tip Efector Final (MECANISM-DE-PREHENSIV, SCULĂ, CAP-DE-FORȚĂ)
	Dacă: Tip Efector Final = MECANISM-DE-PREHENSIV atunci:
Atribut:	Stare Efector Final (ÎNCHIS, DESCHIS)

Fig.5.13. Interfața modulului Controller efector final

Interfața modulului *Controller efector final*, fig.5.13., conține o serie de atribute prin care se realizează un acces la **programul de control** al efecteurului final. Atributele acestui modul au următoarele semnificații:

- Atributul **Număr Identificare** identifică în mod univoc un anumit efecteur final din cadrul unui set de efecteur finali care pot fi cuplați la dispozitivul de ghidare.
- Atributul **Cuplat la Sursa de Energie** este de tip boolean. Prin valoarea ADEVĂRAT sau FALS indică faptul că efecteurul final este cuplat la sursa de energie sau nu.
- Atributul **Tip Efector Final** indică tipul efecteurului final cu care este echipat dispozitivul de ghidare: **MECANISM-DE-PREHENSIV**, **SCULĂ** sau **CAP-DE-FORȚĂ**. Dacă este de tip **MECANISM-DE-PREHENSIV** atunci există și atributul **Stare Efector Final** care poate lua două valori: **ÎNCHIS** sau **DESCHIS**.

MODUL:	CONTROLLER ECHIPAMENT AUXILIAR
Atribut:	Tipul Controlului (LOCAL, DE-LA-DISTANȚĂ)
Atribut:	Cuplat la Sursa de Energie (ADEVARAT, FALS)
Atribut:	Starea de Sincronizare (SINCRONIZAT, NESINCRONIZAT, SINCRONIZARE-ÎN-CURS)
Atribut:	Starea de Funcționare (PORȚIT, OPRIT)

Fig.5.14. Interfața modulului Echipament auxiliar

În fig.5.14. s-a prezentat structura interfeței modulului **Echipament auxiliar**. Atributele acestui modul au următoarele semnificații:

- Atributul **Tipul Controlului** indică faptul că dispozitivul de ghidare, la un anumit moment dat, este sub *comandă locală* sau este permisă *comanda de la distanță*.
- Atributul **Cuplat la Sursa de Energie** este de tip boolean. Prin valoarea ADEVĂ-RAT sau FALS indică faptul că dispozitivul de ghidare este cuplat la sursa de energie sau nu.
- Atributul **Starea de Sincronizare** indică faptul că echipamentul auxiliar, ca o unitate, este sincronizat sau nu. Acest atribut poate lua următoarele valori: SINCRONIZAT, SINCRONIZARE-ÎN-CURS sau NESINCRONIZAT.
- Atributul **Starea de Funcționare** indică prin valorile sale: PORNIT sau OPRIT starea de funcționare a echipamentului auxiliar.

MODUL:	PROCESOR PROGRAM DE APLICAȚIE
Atribut:	Programe de Aplicație Încărcate
Atribut:	Program de Aplicație Selectat
	Dacă: Program de Aplicație Selectat <> NONE atunci:
Atribut:	Stare Execuție Program (GATA-PENTRU-RULARE, RULARE, PAUZĂ, TERMINAT, ABANDONAT)
Atribut:	Cod Eroare
Atribut:	Mod Rulare (RULARE-LIBERĂ, LIMITAT-CICLURI, LIMITAT-PAȘI)
	Dacă: Mod Rulare = LIMITAT-CICLURI atunci:
Atribut:	Număr Cicluri Rămase
	Dacă: Mod Rulare = LIMITAT-PAȘI atunci:
Atribut:	Număr Pași Rămași

Fig.5.15. Interfața modulului Procesor program de aplicație

În fig.5.15. s-a reprezentat structura modulului **Procesor program de aplicație**. Atributele acestui modul au următoarele semnificații:

- Atributul **Programe de Aplicație Încărcate** este o listă cu denumirile programelor de aplicație încărcate în memoria subsistemului de conducere. Dacă nu este nici un program încărcat în memorie atunci atributul are valoarea NONE.
- Atributul **Program de Aplicație Selectat** este de tip șir de caractere și identifică denumirea programului de aplicație selectat sau rulat în mod curent. Dacă nu este nici un program selectat atunci atributul are valoarea NONE. În caz contrar există atributul *Stare Execuție Program*:
 - Atributul **Stare Execuție Program** indică prin valorile sale starea de execuție curentă a programului de aplicație selectat. Înainte de lansare în execuție, un program deja selectat este GATA-PENTRU-RULARE. După lansarea în execuție programul este în stare de RULARE. Dacă execuția este oprită temporar atunci este în starea de PAUZĂ. Dacă s-a executat complet programul atunci se află în starea TERMINAT. Dacă s-a abandonat execuția programului (de exemplu datorită unor erori în execuție atunci se află în starea de ABANDONAT.

- Atributul **Cod Eroare** (Error Code) este de tip întreg și identifică ultima eroare de execuție a programului de aplicație selectat. Valoarea "0" indică faptul că nu s-a înregistrat nici o eroare. Semnificația celorlalte valori este o problemă locală ca și resetarea valorii acestui atribut.
- Atributul **Mod Rulare** (Running Mode) indică modul de desfășurare a execuției programului de aplicație:
 1. Dacă valoarea acestui atribut este RULARE-LIBERĂ (FREE-RUN), atunci un program de aplicație odată lansat în execuție stă în starea RULARE (RUNNING) până la apariția unui eveniment local sau de la distanță, care cauzează oprirea execuției.
 2. Dacă valoarea acestui atribut este LIMITAT-CICLURI (CYCLE-LIMITED) atunci un contor explicit conține numărul de cicluri executate. Atunci când contorul ajunge la zero, rularea programului de aplicație se întrerupe și intră în starea TERMINAT.
 3. Acest atribut poate lua valoarea LIMITAT-PAȘI (STEP-LIMITED) pentru implementări care suportă acest mod. În acest caz, numărul de pași care vor fi executați va fi specificat înainte de lansarea programului. Atunci când numărul de pași, prin decrementare, a ajuns la zero, programul va intra în starea TERMINAT. Acest mod poate fi utilizat pentru depanarea programului de aplicație.
 - Atributul **Număr Cicluri Rămase** (Remaining Cycle Count) este de tip întreg și semnifică numărul de cicluri neexecutate încă. Acest atribut există numai dacă Modul de Rulare este LIMITAT-CICLURI (CYCLE-LIMITED).
 - Atributul **Număr Pași Rămași** (Remaining Step Count) este de tip întreg și semnifică numărul de pași rămași a fi executați atunci când programul este în Modul de Rulare LIMITAT-PAȘI (STEP-LIMITED).

MODUL:	CONTROLER SISTEM ROBOT
Atribut:	Stare Funcționare Robot (INACTIV, ÎNCĂRCAT, READY, ÎN-EXECUȚIE, PAUZĂ-MIȘCARE-ROBOT, NECESITĂ-INTERVENȚIE-MANUALĂ)
Atribut:	Fiecare Resursă Fizică este Cuplată la sursa de energie (ADEVARAT, FALS)
Atribut:	Stare de Sincronizare (NESINCRONIZAT, SINCRONIZARE-ÎN-CURS, SINCRONIZAT)
Atribut:	Tipul Controlului Suportat (COMANDĂ-LOCALA, COMANDA-DE-LA-DISTANȚA)
Atribut:	Capacitatea de Funcționare (OPERAȚIONAL, PARȚIAL-OPERAȚIONAL, NEFUNCȚIONAL, NECESITĂ-DISPOZIȚIE)

Fig.5.16. Interfața modulului Controler sistem robot

În fig.5.16. s-a prezentat interfața modulului **Controler sistem robot**. Atributele acestui modul au următoarele semnificații:

- Atributul **Stare Funcționare Robot** (Robot Operation State) indică starea de funcționare a sistemului robot. Stările de funcționare posibile ale sistemului robot

(INACTIV, ÎNCĂRCAT, READY, ÎN-EXECUȚIE, PAUZĂ-MIȘCARE-ROBOT, NECESITĂ-INTERVENȚIE-MANUALĂ) au fost descrise în paragraful 5.3.4.

- Atributul **Fiecare Resursă Fizică este Cuplată la sursa de energie** (Any Physical Resource Power On), este un atribut de tip boolean, care prin valorile de ADEVĂRAT sau FALS indică starea de alimentare (sau lipsa de alimentare) cu energie a sistemului robot.
- Atributul **Stare de Sincronizare** indică prin valorile sale starea de sincronizare a întregului sistem robot. Stările posibile (NESINCRONIZAT, SINCRONIZARE-ÎN-CURS, SINCRONIZAT) au fost descrise în paragraful 5.3.3.
- Atributul **Tipul Controlului Suportat** indică prin valorile sale faptul că un "agent local" posedă controlul asupra resurselor fizice ale sistemului (COMANDĂ-LOCALĂ) sau este posibil controlul acestor resurse de la distanță (COMANDĂ-DE-LA-DISTANȚĂ). Posesia controlului reprezintă capacitatea de a efectua acțiuni care schimbă valoarea unui atribut sau a mai multor atribute ale modulelor. Comanda locală poate reprezenta acțiunea unui operator uman sau a unei proceduri automate. Oricare dintre ele împiedică comanda de la distanță.
- Atributul **Capacitatea de Funcționare** indică prin valorile sale starea resursei cu aceeași denumire. Stările posibile ale acestei resurse (OPERAȚIONAL, PARȚIAL-OPERAȚIONAL, NEFUNCȚIONAL, NECESITĂ-DISPOZIȚIE) au fost descrise în paragraful 5.3.1.

5.5. Concluzii

În acest capitol s-a propus un nou model de accesare de la distanță a resurselor fizice și logice ale roboților industriali. Noutatea modelului constă în faptul că are la bază **concepția arhitecturii deschise**, prezentată în capitolul 4. Astfel, în mod asemănător echipamentelor CNC, se propus și pentru controlere de roboți industriali dezvoltarea unei arhitecturi deschise, independente de producător, caracterizată printr-o structură modulară a software-ului de sistem. Fiecare "modul" controlează una sau mai multe resurse ale robotului industrial. Astfel accesul de la distanță la controlul resurselor se reduce la accesul standard la aceste module software.

Înainte de elaborarea acestui model, s-au stabilit acele resurse fizice și logice ale unui robot industrial (integrat într-un Hipersistem CIM) care pot fi sau în general sunt necesare să fie accesate prin comenzi de la distanță. Astfel a fost elaborat **un model al resurselor fizice și un model al resurselor logice** ale roboților industriali. Prin aceste modele s-a realizat o descriere generală a resurselor fizice și logice amintite, propunând o nouă interpretare a acestora. De asemenea au fost stabilite principalele module software, aferente controlului resurselor respective.

La stabilirea modulelor respectiv a funcțiilor standarde ale acestora, s-a luat în considerare doar cerințele generale ale realizării comenzii de la distanță a robotului industrial. Funcțiile modulelor au fost descrise printr-o serie de "atribute". Modelul elaborat propune un acces simplu la atribute, prin servicii de citire (READ) respectiv de scriere (WRITE). Prin aceste servicii se pot obține informații despre starea curentă de funcționare a unei anumite resurse (prin citirea valorii curente a unui atribut) sau se poate realiza controlul resursei respective (prin modificarea prin scriere a va-

lorii atributului). Astfel se poate realiza atât monitorizarea funcționării cât și comanda de la distanță a unui robot industrial. Pe lângă serviciile de citire și scriere a fost prevăzut suplimentar și o serie de servicii legate de încărcarea, ștergerea, selectarea, lansarea în execuție etc. a programelor de aplicație robot.

Trebuie arătat că prin modelul de accesare elaborat și prezentat în acest capitol nu s-a propus rezolvarea tuturor problemelor tehnice legate de realizarea comenzii de la distanță a roboților industriali.

Meritul acestui model constă în faptul că încearcă pentru prima oară tratarea problemei **comenzii "deschise" de la distanță** a unui robot industrial în paralel cu problema **arhitecturii deschise a controlerului de robot**.

CAPITOLUL 6.

ELABORAREA UNOR STRUCTURI DE SCHISE DE COMANDĂ PENTRU UN MINI-SISTEM CIM

6.1. Introducere

În capitolele anterioare s-au prezentat cercetările existente pe plan mondial, referitor la elaborarea și introducerea în Hipersisteme CIM a structurilor deschise de comandă. În concordanță cu aceste cercetări, s-a încercat elaborarea unor structuri deschise de comandă, pentru un mini-sistem CIM existent la Universitatea din Oradea (CIM2000). În acest capitol se prezintă câteva dintre structurile elaborate.

Trebuie menționat faptul că, pe lângă elaborarea unui proiect de introducere a unor structuri deschise de comandă, s-a pus accent mai mult pe dezvoltarea câtorva structuri de comandă de la distanță a robotului industrial RV-M1, din cadrul acestui mini-sistem.

6.2. Prezentarea mini-sistemului CIM2000

CIM2000 este un sistem de fabricație integrat cu calculatorul, realizat în scop educațional de către firma DEGEM System Ltd. din Israel.

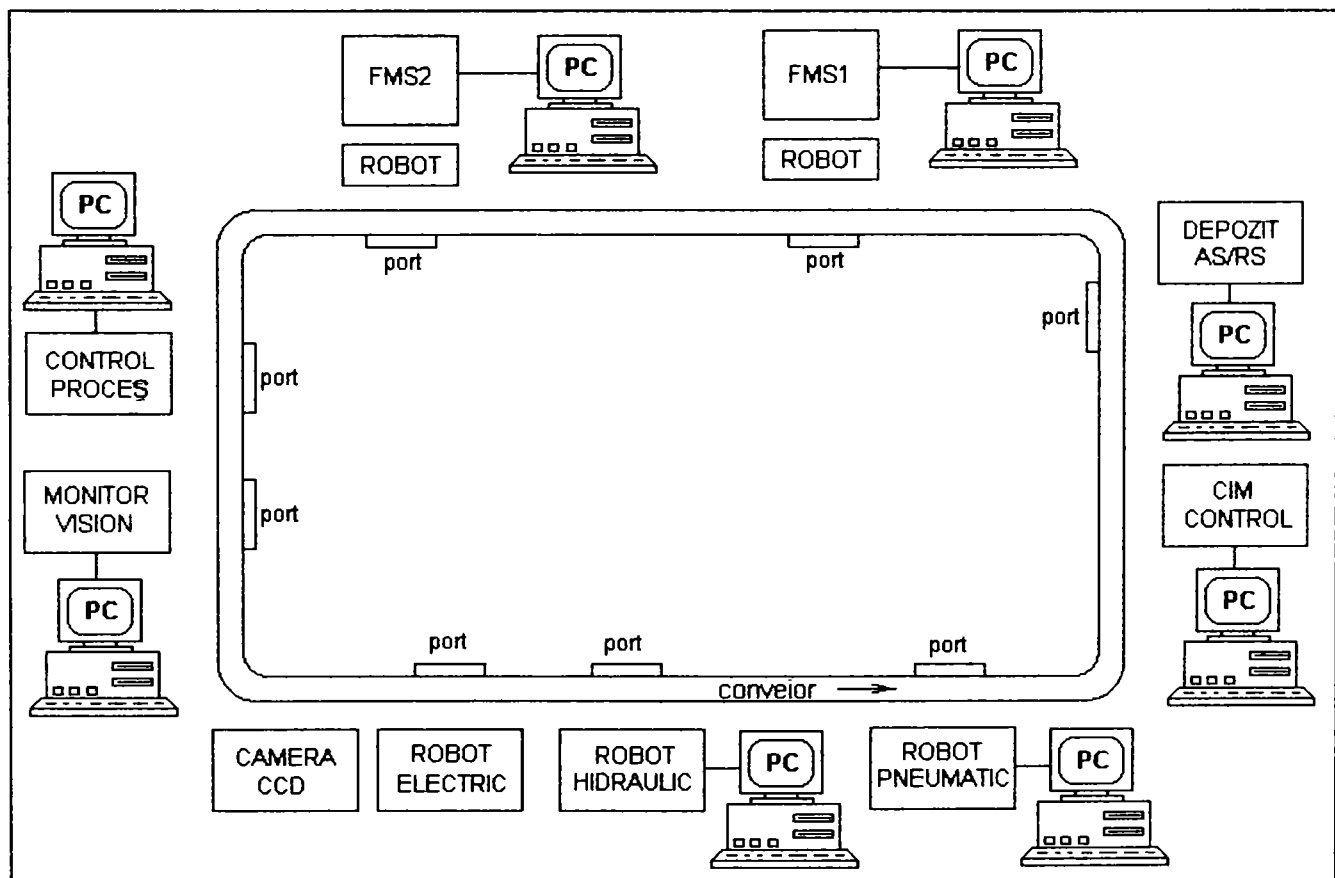


Fig.6.1. Structura sistemului complet CIM2000, [24]

Varianta completă (fig.6.1.) conține două module (sisteme) de fabricație: unul pentru prelucrarea pieselor cilindrice (FMS1 sau FMS2101 - conceput în jurul unui strung cu comandă numerică) iar celălalt, pentru prelucrarea pieselor prismatice (FMS2 sau FMS2102 – conceput în jurul unei mașini de frezat cu comandă numerică), o stație pentru tratamente termice (Process Control); o stație de control dimensional care utilizează în acest scop o cameră CCD (Vision Monitor – VI2000); o stație de asamblare mecanică, realizată prin intermediul unei prese respectiv unor manipuloare hidraulice (Hidraulic Robot); o stație de sortare și încărcare semifabricate, realizată prin intermediul unor manipuloare pneumatice (Pneumatic "Robot"-PN2800); o stație de depozitare și regăsire automată (AS/RS – ST2000); un sistem de transport de tip conveyer respectiv o stație de control central (CIM-Control).

La Universitatea din Oradea există o **variantă redusă** a sistemului CIM2000 descris mai sus.

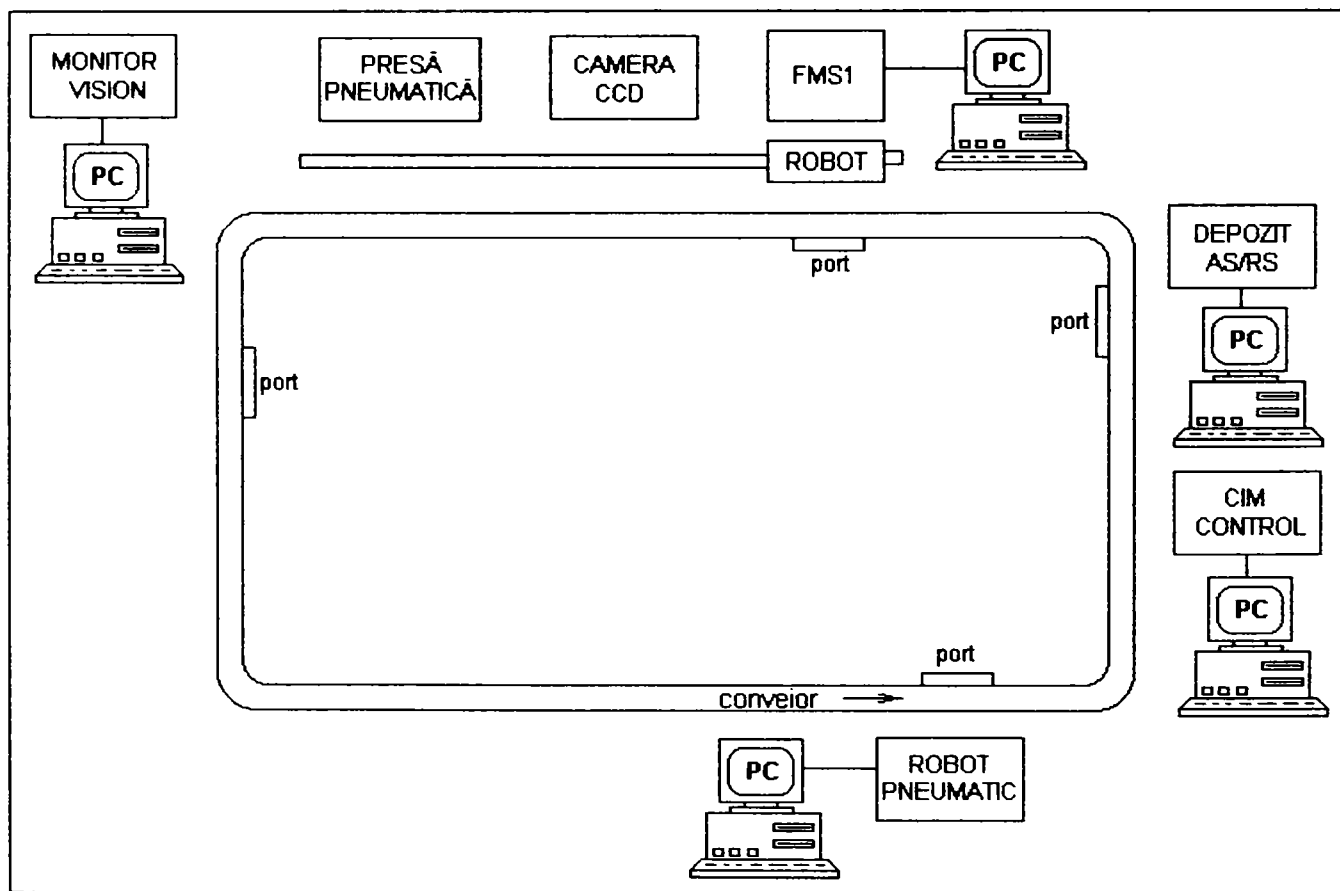


Fig.6.2. Structura mini-sistemului CIM2000, [11]

Numit **Mini-sistem CIM2000** (fig.6.2.), acesta conține doar sistemul de fabricație flexibilă FMS1 (Flexible Manufacturing System) în care a fost integrată (fizic și informațional) stația de asamblare (Pneumatic Press) respectiv stația de control dimensional (VI2000 – camera CCD); respectiv conține stația PN2800 de sortare și încărcare semifabricate (Pneumatic "Robot"), stația ST2000 de depozitare și de regăsire automată (AS/RS – Automated Storage and Retrieval System) și stația de control central (CIM-Control).

Mini-sistemul este dotat cu un singur robot industrial (RV-M1) pentru deservirea stațiilor FMS1, de control dimensional VI2000 și de asamblare (Pneumatic Press).

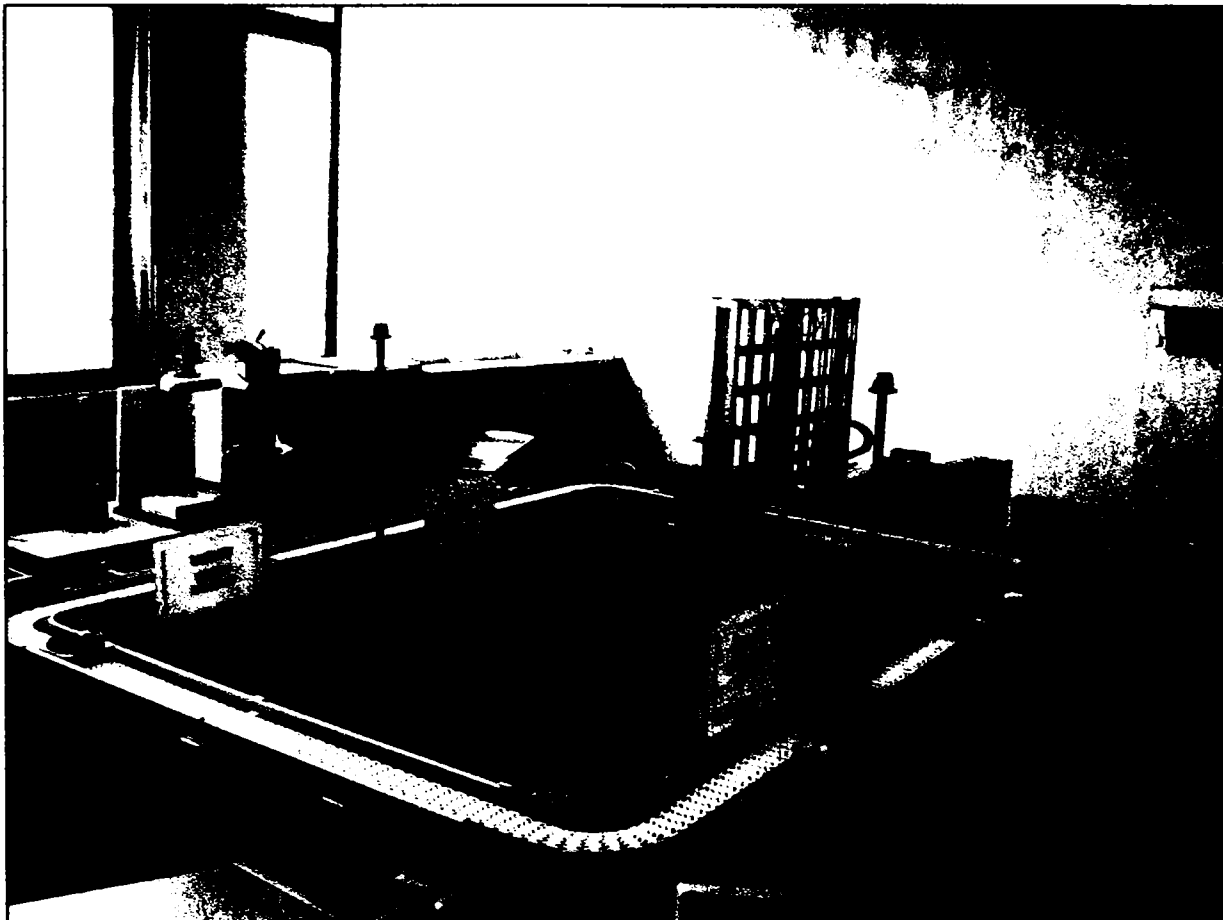


Fig.6.3. Mini-sistemul CIM2000 de la Universitatea din Oradea - vedere generală



Fig.6.4. Robotul RV-M1, strungul NCL2000, camera CCD, presa pneumatică

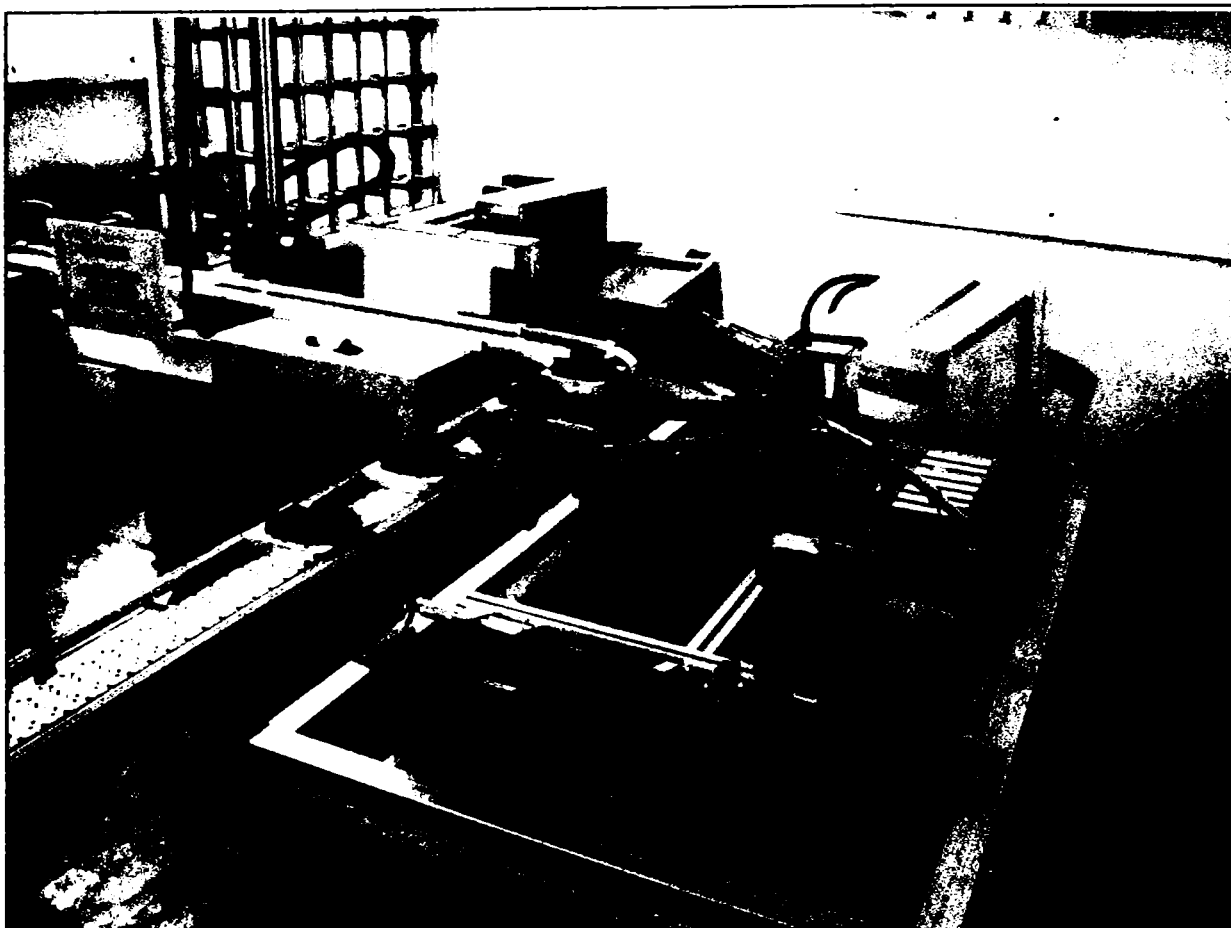


Fig.6.5. Stația ST2000 de depozitare și de regăsire automată respectiv stația PN2800 de sortare și încărcare semifabricate

În figurile.6.3.-6.5. s-a prezentat câte o imagine de ansamblu a mini-sistemului CIM2000, de la Universitatea din Oradea. Poate fi identificat ușor: sistemul de transport de tip conveior, sistemul de fabricație flexibilă (FMS1): cu strungul NCL2000 și robotul industrial RV-M1; stația VI2000 cu camera CCD; stația ST2000 de depozitare și de regăsire automată respectiv stația PN2800 de sortare și încărcare semifabricate.

6.2.1.Structura detaliată a mini-sistemului

În fig.6.6. s-a prezentat schema de amplasare detaliată a mini-sistemului CIM2000. Astfel poate fi pusă în evidență structura stațiilor din cadrul sistemului FMS1 (FMS2101) care sunt servite prin intermediul robotului industrial RV-M1 (montat pe sistemul de transport local numit: Slide).

De asemenea, pot fi evidențiate structurile stațiilor PN2800 (DSC – Depozit Semifabricate Cilindrice, DSP – Depozit Semifabricate Prismatice, DP – Depozit Palete de transport, MP – Manipulator pneumatic de Palete, MR – Manipulator pneumatic de semifabricate Cilindrice, CO – Container pentru eliminarea rebuturilor) respectiv ST2000 (MP – Manipulator pneumatic de Palete, Stecker – Manipulator electro-pneumatic de palete care servește depozitul - are funcția unui robot de depozit, DPF – Depozit pentru Piese Finite și piese neterminat – organizat sub formă de rafturi verticale).

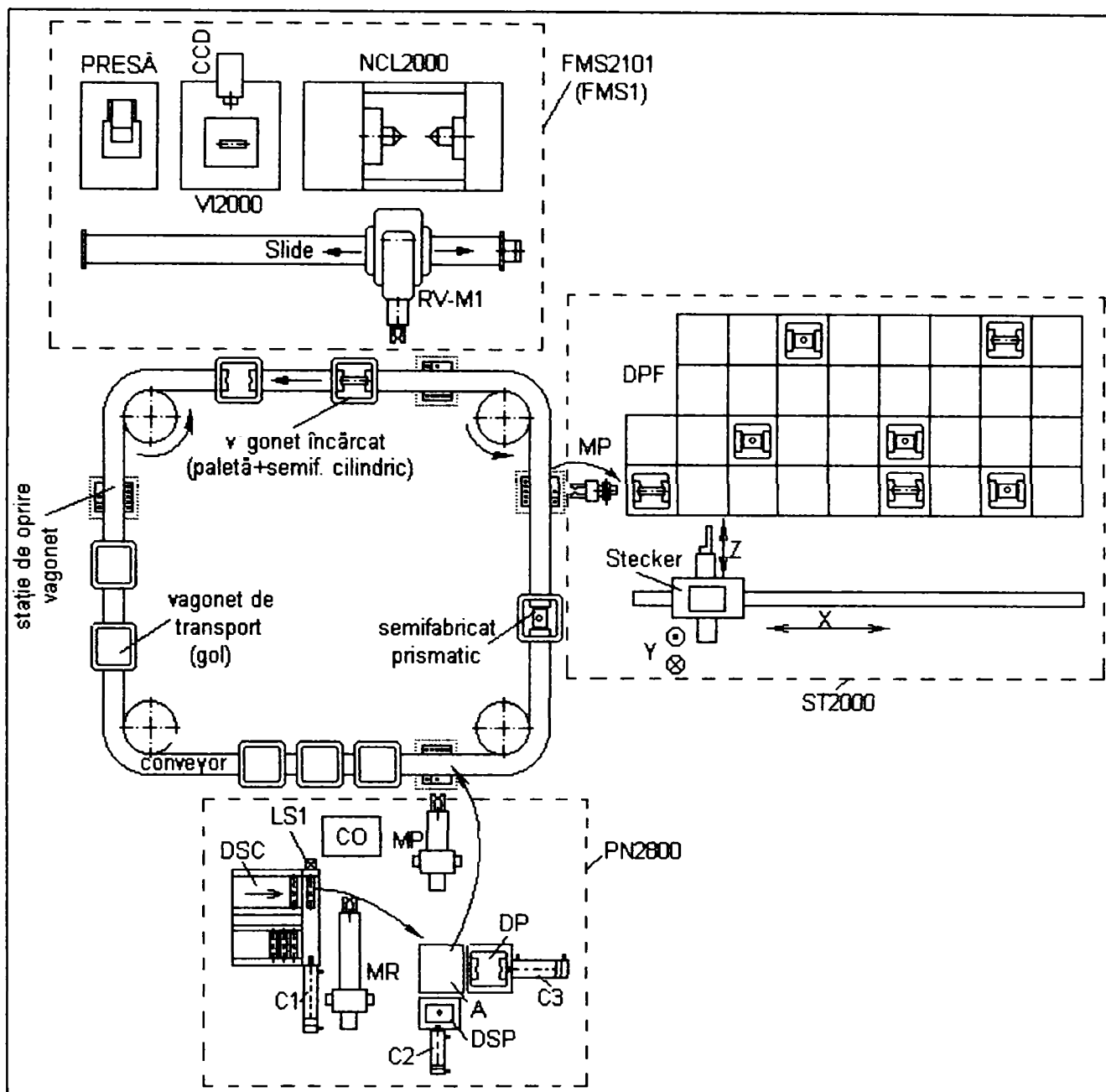
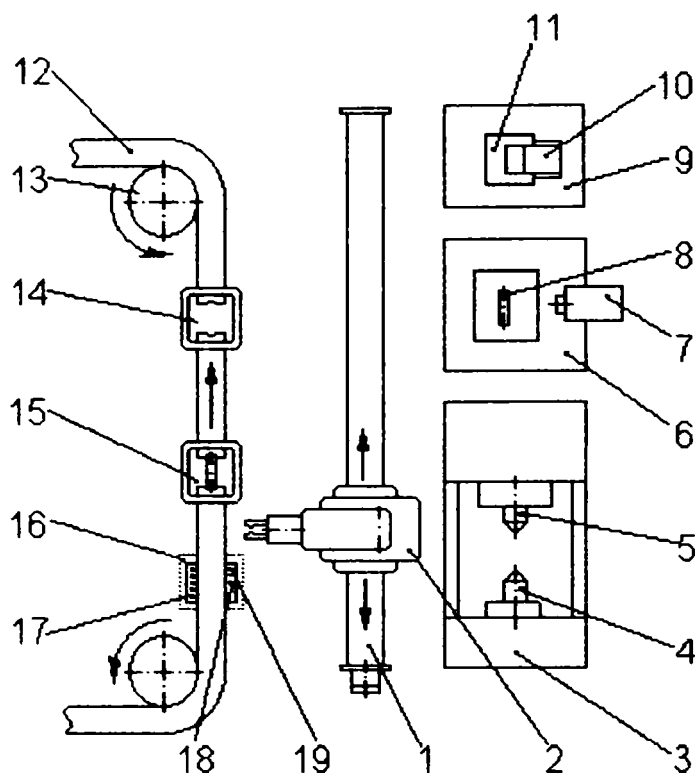


Fig.6.6. Schema de amplasare detaliată a mini-sistemului CIM2000, [11]

Din cele prezentate mai sus, mini-sistemul CIM2000 reprezintă o structură minimală, necesară pentru un sistem de fabricație integrat cu calculatorul sub genericul CIM. Astfel pot fi regăsite componentele fizice, esențiale ale unui sistem CIM: sistem de fabricație flexibilă (FMS – Flexible Manufacturing System), sistem de asamblare flexibilă (FAS – Flexible Assembling System), sistem de control și asigurare a calității (CAQA – Computer Aided Quality Assurance), sistem de depozitare și regăsire automată, (AS/RS – Automated Storage/Retrieval System), sistem de sortare și alimentare cu semifabricate, sistem de transport uzinal (de obicei AGVS – Automated Guided Vehicle System, în cazul de față înlocuit cu un sistem de transport de tip conveyor).

Însă dorința de obținere a unui sistem CIM cât mai redus (adică cât mai ieftin) a condus atât la reorganizarea componentelor de fabricație (prelucrare), de asamblare și de control al calității într-un singur sistem FMS1 cât și la necesitatea simplificării structurii informaționale.

Structura detaliată a sistemului de fabricație flexibilă FMS2101, [5]



1. Sistemul de transport Slide
2. Robot industrial
3. Stația de prelucrare prin strunjire
4. Vârful mobil din păpușă
5. Vârful fix din universal
6. Stație de control dimensional
7. Cameră CCD
8. Piesă sub cameră CCD
9. Stația de asamblare
10. Presă pneumatică
11. Masa preseii
12. Conveior cu bandă
13. Rolă pentru întinderea benzii
14. Vagonet gol
15. Vagonet încărcat (cu piesă)
16. Stație de oprire vagonet (buffer)
17. Senzori de identificare vagonet
18. Sensor de prezență vagonet
19. Opritor mecanic

Fig.6.7.

Sistemul de fabricație FMS2101 (fig.6.7.), conține o **stație de prelucrare prin strunjire** (NCL 2000), o **stație de control dimensional** (VISION 2000), o **stație de asamblare** respectiv **sistemul micro-robot RV-M1**. Alimentarea cu piese a sistemului se realizează prin intermediul unui conveior (12). Pe acest sistem de transport, piesele (semifabricatele și produsele finite) sunt transportate pe vagoaneți iar descărcarea/încărcarea vagoanelor are loc numai în stațiile de oprire vagonet.

Stația de prelucrare prin strunjire (3) este formată dintr-un strung CNC. Semifabricatele sunt fixate între vârfuluri 4 și 5, astfel încât până când vârful din universal este fix, vârful din păpușă strungului este acționat de un cilindru pneumatic. Programele de prelucrare sunt stocate într-un calculator pe rol de mini-echipament de comandă numerică. Selectarea programului de prelucrare se realizează de către operator, la pornirea sistemului sau automat prin recepționarea, de la echipamentul de comandă al robotului industrial RV-M1, a codului binar corespunzător numărului programului preselectat.

Stația de control dimensional (6) conține o cameră CCD prin intermediul căreia este preluată imaginea piesei și un calculator personal PC care realizează compararea numerică a imaginii piesei măsurate cu imaginea memorată a unei piese de referință. Rezultatul comparării (piesă bună sau piesă rebut) este transmis, prin intermediul unei interfețe I/O, către echipamentul de comandă al robotului. Controlul dimensional al unei piese așezate, prin intermediul robotului RV-M1, sub camera CCD începe automat la recepționarea de către stație a unei cereri de control dimensional.

Stația de asamblare (9) conține o presă pneumatică care se utilizează pentru realizarea unor operații de asamblare prin presare. În configurația actuală, stația de

asamblare este utilizată pentru realizarea unui produs finit prin asamblarea unei piese prismatice cu o piesă cilindrică prelucrată.

Sistemul micro-robot (format din robotul industrial RV-M1 și sistemul de transport Slide) se utilizează pentru servirea stațiilor amintite, respectiv pentru descărcarea și încărcarea vagonetelor sosiți în stația de oprire (16). Transportul pieselor între stațiile 3, 6, 9 și 16 este realizat prin deplasarea robotului pe Slide în pozițiile memorate din fața acestor stații. Între echipamentul de comandă al robotului RV-M1 și sistemele de conducere nemijlocită ale stațiilor amintite există o comunicație prin intermediul unor semnale discrete. În acest mod echipamentul de comandă al robotului RV-M1, realizează și coordonarea respectiv supravegherea succesiunii operațiilor din cadrul sistemului flexibil FMS2101.

6.2.2. Arhitectura sistemului de comandă

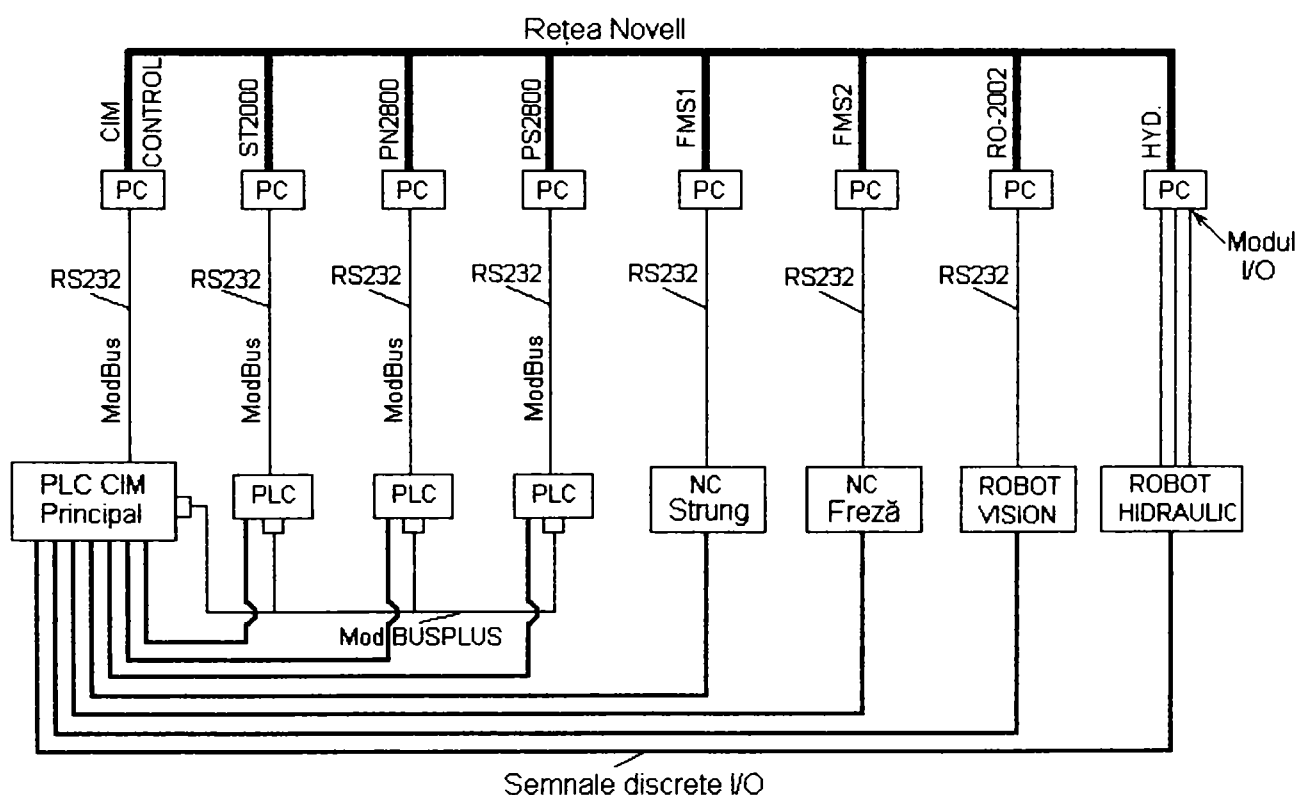


Fig.6.8. Structura informațională de principiu a sistemului complet CIM2000, [24]

În fig.6.8. s-a evidențiat existența în cadrul sistemului complet a 4 tipuri de comunicație:

- comunicația serială (RS232) între calculatoarele PC și automatele programabile PLC (protocolul ModBus) respectiv între calculatoarele PC și echipamentele de comandă numerică din cadrul FMS1, FMS2 și RO2002;
- comunicația între automatele programabile PLC, legate într-o rețea de comunicație "master-slave" (protocolul ModBusPlus);
- comunicația prin intermediul unor semnale numerice discrete (discret signal I/O);
- comunicația între calculatoarele PC prin intermediul unei rețele locale de tip Novell.

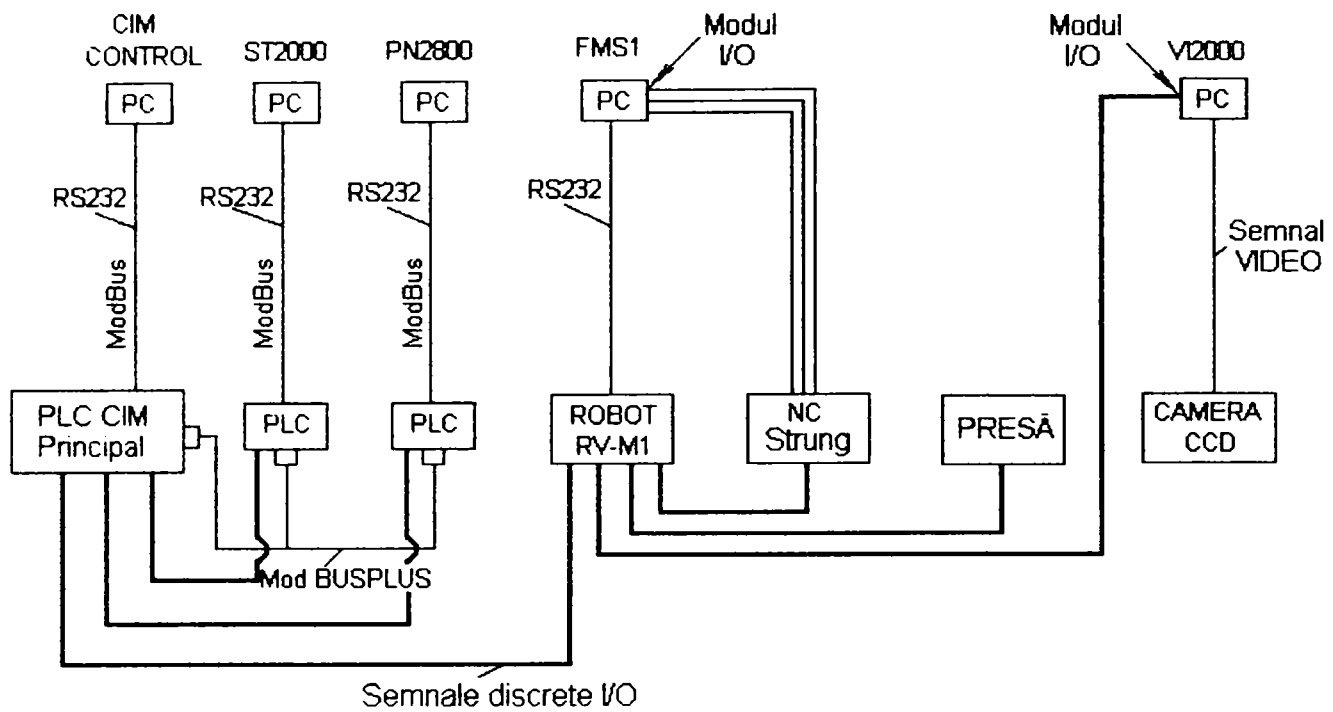


Fig.6.9. Structura informațională de principiu a mini-sistemului CIM2000, [7]

În fig.6.9. s-a prezentat structura informațională de principiu a mini-sistemului CIM2000. Se poate observa diferența esențială față de structura informațională a sistemului complet, adică lipsa comunicațiilor între calculatoarele PC (lipsa rețelei locale de tip Novell). Ca urmare, coordonarea procesului de fabricație din FMS1 respectiv din VI2000 cu procesele din cadrul stațiilor PN2800, ST2000 și CIM-Control se realizează numai prin intermediul semnalelor discrete I/O.

Pentru punerea în evidență a acestui aspect, s-au întocmit figurile 6.10. și 6.11. Se poate observa că posibilitatea schimbului de "date" între stații se rezumă doar la schimburi de date între stațiile PN2800, ST2000 și CIM-Control (prin intermediul rețelei de comunicație ModBusPlus). Liniile I/O sunt utilizate doar pentru sincronizarea activităților din cadrul stațiilor amintite cu stația de control central CIM-Control.

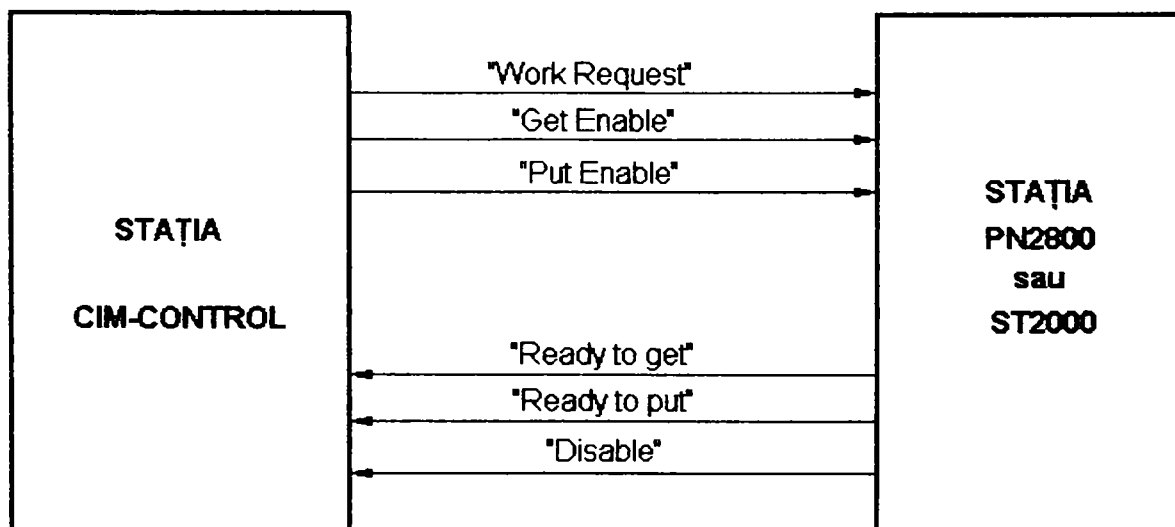
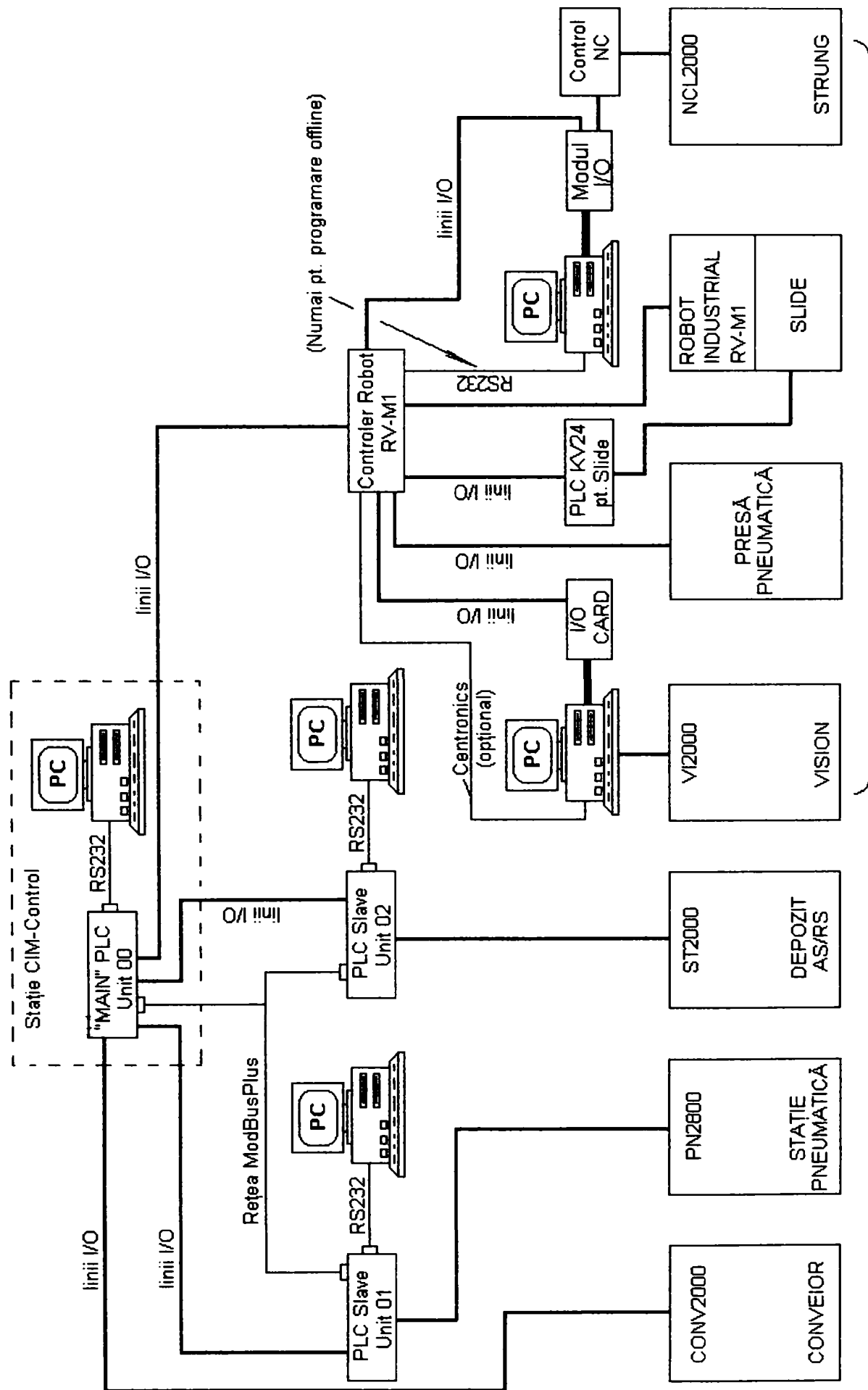


Fig.6.10. Comunicația între stația CIM-Control și stațiile PN2800 și ST2000, [24]



FMS2101 - Sistem Flexibil de Fabricație

Fig.6.11. Structura informațională detaliată a mini-sistemului CIM2000, [3,7]

În fig.6.10. s-a prezentat legătura informațională a stației CIM-Control cu stațiile PN2800 și respectiv ST2000, la nivel de semnale discrete I/O. Pe baza figurii, stația CIM-Control emite semnalul "Work request" către stația PN2800 sau ST2000 prin care se autorizează funcționarea acestora în regim automat. La această "cerere de lucru" stațiile răspund cu "Ready to get" ("Sunt gata pentru preluarea unui semifabricat/piesă finită") și/sau "Ready to put" ("Sunt gata pentru eliberarea unui semifabricat").

Prin aceste semnale, stațiile amintite își exprimă disponibilitatea lor pentru preluarea și/sau așezarea unui semifabricat/piesă finită de pe sau pe conveyior. Ciclurile de introducere (preluare) sau de scoatere (așezare) sunt autorizate (adică lansate) prin unul din semnalele: "Get Enable" ("Autorizare preluare") sau "Put Enable" ("Autorizare așezare") emise de stația CIM-Control. Pe parcursul ciclurilor de introducere și de scoatere, stațiile PN2800 și ST2000 emit semnalul "Disable" (adică "Ocupat").

Schimbul de informații între stațiile respective și stația CIM-Control, cu privire la gestiunea semifabricatelor/pieselor finite (prin intermediul unui "cod articol"), se realizează prin rețeaua de automate programabile: MODBUS Plus, fig.6.11. Urmărirea fluxului semifabricatelor/pieselor în cadrul mini-sistemului CIM2000 este realizată prin intermediul stației CIM-Control. Aceasta urmărește în acest scop, prin intermediul semnalelor discrete I/O și respectiv a senzorilor de identificare vagonet, fluxul și conținutul vagonetelor de transport de pe Conveyior.

În fig.6.12. s-au prezentat legăturile informaționale (prin intermediul semnalelor discrete I/O) ale echipamentului de comandă al robotului industrial RV-M1. În cadrul figurii sunt prezentate atât legăturile informaționale cu stațiile deservite din cadrul sistemului flexibil FMS2101 (NCL2000, VI2000, stația de asamblare) cât și cu controler-ul Slide-ului respectiv cu automatul programabil din cadrul stației CIM-Control.

Schimbul de informații între echipamentul de comandă al robotului RV-M1 și **automatul programabil CIM-Control** se rezumă la informațiile necesare sincronizării operațiilor de alimentare a sistemului FMS2101 cu semifabricate (cilindrice și prismatice) precum și la preluarea codului binar (pe 3 biți) al numărului programului de prelucrare. Acest număr apoi este transmis (tot prin intermediul a 3 semnale discrete) către echipamentul de comandă numerică al strungului, unde este utilizat pentru selectarea automată a programului de prelucrare.

Numărul poziției de pe Slide, în care se dorește deplasarea robotului se specifică tot în cod binar (prin intermediul a 3 semnale discrete care formează "adresa de poziționare"). Mișcarea începe la recepționarea de către controlerul Slide-ului a unei cereri de poziționare, iar sfârșitul deplasării este confirmat prin semnalul "Slide în poziție".

În cazul stației de prelucrare, comenzile emise de echipamentul de comandă al robotului RV-M1 se referă la fixarea semifabricatului între vârfuri și la autorizarea începerii prelucrării. La terminarea prelucrării este emis un semnal de "sfârșit prelucrare". Înainte de scoaterea piesei prelucrate din stație, este verificată starea închisă sau deschisă a ușii strungului.

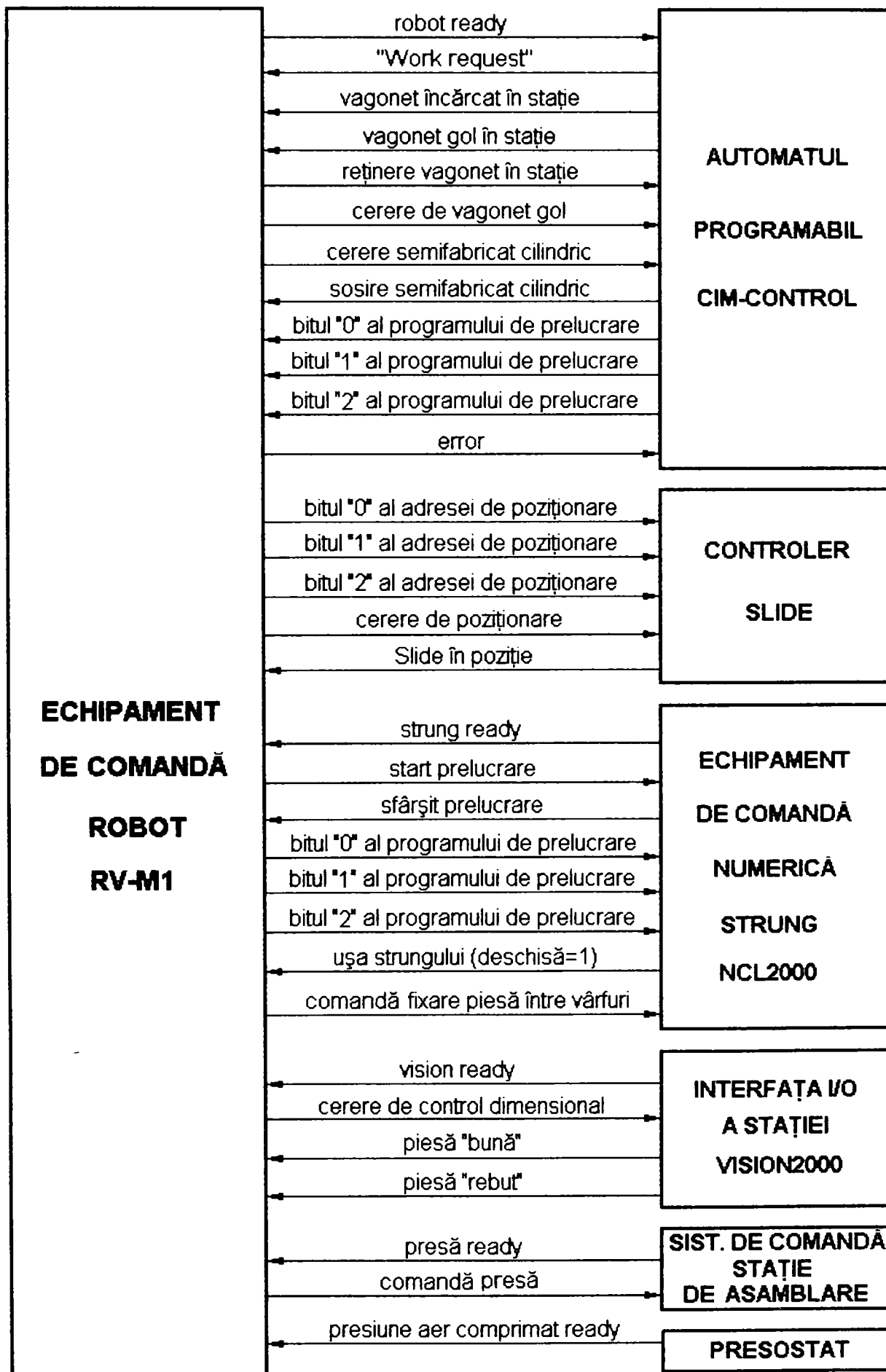


Fig.6.12. Legăturile informaționale ale robotului RV-M1, [3, 5]

În cazul stației VISION2000 nu se confirmă terminarea controlului dimensional. Astfel sfârșitul "comparării" se detectează din stările logice ale semnalelor: "piesă bună" și "piesă rebut". Adică în timpul controlului ambele semnale au nivele logice "0". La terminarea controlului, ori semnalul de "piesă bună" ori cel de "piesă rebut" va avea nivelul logic "1".

În cazul stației de asamblare există o singură confirmare privind faptul că stația este pregătită pentru o nouă operație și o singură comandă prin care se lansează operația de presare. Având în vedere că, atât în cazul fixării semifabricatului între vârfuri (la stația de prelucrare) cât și în cazul stației de asamblare, foarte important este ca presiunea aerului comprimat să nu scadă sub o anumită valoare minimă, sistemul este prevăzut cu un presostat (releu de presiune).

6.3. Elaborarea unui proiect de introducere în mini-sistemul CIM2000 a unor structuri deschise de comandă

Așa cum a fost prezentat în cadrul paragrafului 6.2.2., schimbul de informații între stațiile: CIM-Control, VI2000, NCL2000 și robotul RV-M1 se rezumă la **sincronizarea operațiilor prin intermediul semnalelor discrete I/O**. Ca urmare, între aceste stații nu pot exista schimburi de informații la nivel de "mesaje", adică nu pot fi emise "comenzi" privind transferul, selectarea și lansarea programelor de fabricație, respectiv pentru urmărirea și/sau modificarea on-line a stării de funcționare a unei stații. În acest mod, **posibilitățile de programare, urmărire și optimizare a fabricației**, la nivelul întregului mini-sistem CIM2000 **sunt extrem de reduse**, [7]. Aceasta afectează desigur flexibilitatea de funcționare a întregului mini-sistem.

Pentru înlăturarea acestor inconveniente, **a devenit necesară modernizarea structurii de comandă** a întregului mini-sistem CIM2000. Toate modificările propuse a fi realizate, au fost incluse într-un **proiect de modernizare**. Acest proiect a prevăzut introducerea în mini-sistemul CIM2000, pe cât este posibil, a unor structuri deschise de comandă, [4].

6.3.1. Prezentarea proiectului

La baza elaborării proiectului de modernizare a structurii de comandă a mini-sistemului CIM2000, au stat structurile tipice de automatizare a fabricației din cadrul Hipersistemelor CIM, care au fost prezentate în capitolele 2 și 3. De asemenea s-au ținut cont de următoarele reguli generale:

1. Structura sistemului de comandă a Hipersistemului CIM, la nivelul compartimentelor de execuție (hala industrială), este formată în general dintr-o **rețea coloană de fabrică** (Factory Backbone Network) și respectiv dintr-un număr de **rețele de celulă** (Cell Network) din cadrul celulelor de fabricație (Manufacturing Cell).
2. La rețeaua de celulă, se conectează fiecare echipament de fabricație din cadrul unei celule de fabricație. Această conectare poate să fie realizată direct, dacă echipamentele respective sunt sau pot fi dotate cu interfețe de tip LAN, sau pot fi conectate indirect, prin intermediul unui "calculator de interfață", care la rândul său este dotat cu această interfață LAN.

3. Comanda și monitorizarea funcționării unei celule de fabricație se realizează prin intermediul unui calculator care are funcția de **Controler de Celulă** și de asemenea este dotat cu interfață LAN.
4. Rețelele de celulă sunt conectate la rețeaua coloană de fabrică, fie prin intermediul Controlerelor de Celulă, fie prin intermediul echipamentelor de tip **Poartă** (Gateway), fie prin echipamentelor de tip **Punte** (Bridge).
5. Tot la rețeaua coloană de fabrică sunt conectate în general și echipamentele de comandă ale sistemelor de depozitare și regăsire automată AS/RS (Automated Storage and Retrieval System), respectiv ale sistemului de transport uzinal (în general sistem AGV).
6. În funcție de mărimea și complexitatea sistemului de producție, acesta din urmă poate avea o structură care prevede integrarea celulelor de fabricație în ateliere și/sau secții. În aceste cazuri, comanda și monitorizarea activităților de fabricație din cadrul unui atelier sau secții (formate din mai multe celule de fabricație) se realizează prin intermediul unor calculatoare pe post de **Controler de Local** (Area Controller). Controlerile Locale sunt conectate de asemenea la rețeaua coloană de fabrică.
7. Comanda și monitorizarea întregului proces de fabricație, la nivelul halei industriale, se realizează de către un calculator pe post de **Controler Principal de Uzină** (Host Plant Controller) care este conectat la rețeaua coloană de fabrică.
8. Atât pentru rețeaua coloană de fabrică, cât și pentru rețelele de celulă pot fi utilizate diferite metode de transmitere a comenzilor. Una dintre cele mai ieftine este **metoda CSMA/CD** adică cea de tip "Ethernet" (IEEE802.3).

În urma analizei structurii existente de comandă a mini-sistemului CIM2000, precum și ținând cont de:

- necesitatea introducerii unor structuri deschise de comandă;
- și **posibilitățile financiare ale universității**

s-a ajuns la un **proiect de modernizare** care prevede realizarea în timp a structurii de comandă prezentată în fig.6.13.

Pe baza acestei figuri, pentru **asigurarea unei arhitecturi deschise de comandă**, toate echipamentele din cadrul mini-sistemului vor fi interconectate prin intermediul unei rețele locale LAN. De asemenea, fiecare echipament de comandă, care nu poate fi dotat (din motive constructive) cu interfețe LAN, vor fi interfațate cu calculatoare PC. Aceste calculatoare PC, având la bază o arhitectură hardware deschisă, vor putea fi dotate cu interfețe LAN adecvate pentru interconectarea deschisă a tuturor echipamentelor.

Întâi vor fi interconectate (prin intermediul unei rețele locale de tip LAN) echipamentele din cadrul sistemului flexibil de fabricație FMS2101. Având în vedere numărul redus al componentelor, acesta va funcționa în continuare sub forma unei celule de fabricație (manufacturing cell). Ca urmare rețeaua LAN va avea rolul unei "Rețele de celulă" (Cell Network). La această rețea, stațiile VI2000 (de control dimensional) respectiv NCL2000 (de prelucrare prin strunjire) vor fi conectate prin intermediul calculatoarelor existente (care vor fi dotate cu interfețe LAN).

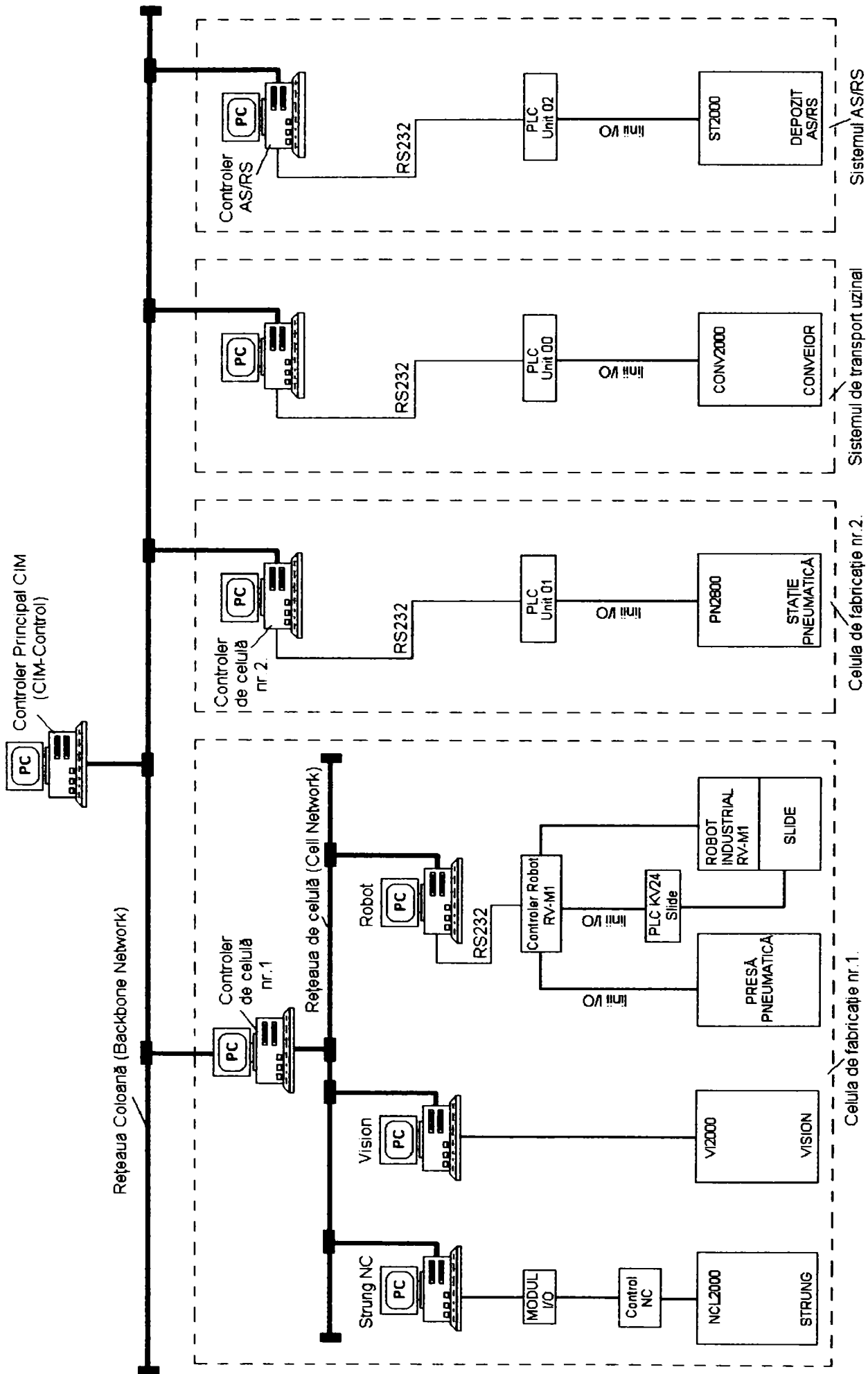


Fig.6.13. Structura sistemului de comandă propusă în cadrul proiectului de modernizare, [12]

Luând în considerare faptul că echipamentul de comandă al robotului industrial RV-M1 nu dispune de interfață LAN și sunt șanse puține (d.p.d.v. tehnic) pentru dotarea echipamentului cu o interfață dedicată, acesta va fi conectat la rețeaua de celulă prin intermediul unui **calculator PC suplimentar** (inexistent în momentul întocmirii proiectului) dotat cu interfață LAN.

În cadrul proiectului s-a considerat că Slide-ul trebuie să rămână sub comanda exclusivă a controler-ului de robot RV-M1, și deci nu este necesară conectarea acestuia la rețeaua de celulă. În caz contrar ar fi necesară încă un calculator PC în plus, ceea ce nu poate fi justificat financiar. S-a ajuns la un compromis identic și în cazul preseii pneumatice (comanda unui singur motor pneumatic linear nu necesită supervizare prin intermediul unui calculator PC). Astfel atât Slide-ul cât și presa pneumatică rămân sub comanda exclusivă a controler-ului de robot RV-M1.

Pentru comanda și monitorizarea activităților de fabricație din cadrul celulei (nr.1.), va fi integrat în aceasta, în mod suplimentar un calculator PC. Acesta din urmă va fi conectat, prin intermediul unei interfețe LAN, la rețeaua de celulă și va avea rol de **controler de celulă** (cell controller). Prin intermediul acestui controler de celulă va fi realizată interconectarea rețelei de celulă cu o rețea LAN, care va străbate (sub forma unei coloane vertebrale) întregul mini-sistem CIM2000, și care astfel va avea denumirea: **rețeaua coloană**.

La rețeaua coloană (backbone network) vor fi conectate în continuare (prin intermediul a câte unei interfețe LAN) :

1. calculatorul de comandă a stației pneumatice PN2800 (aceasta din urmă va fi a doua celulă de fabricație, iar calculatorul PC va fi pe rol de **controler de celulă – nr.2.**);
2. calculatorul de comandă a stației ST2000 de depozitare și de regăsire automată care va fi pe rol de **controler AS/RS**;
3. și respectiv un calculator suplimentar, având rolul supravegherii (împreună cu fostul automat programabil "central": PLC Unit00) **transportului "uzinal"** (adică a fluxului semifabricatelor/pieselor finite pe conveior).

Tot la rețeaua coloană va fi conectat și calculatorul de comandă respectiv de monitorizare a întregului mini-sistem CIM2000 (fostul calculator "CIM-Control") care va fi numit în continuare: **Controler Principal CIM**.

Având în vedere numărul relativ redus al "celulelor de fabricație" din cadrul mini-sistemului CIM2000, nu s-a putut justifica introducerea unui nou calculator PC pe rol de Controler Local (Area Controller).

Pentru interconectările LAN din figură s-a prevăzut **utilizarea rețelelor de tip Ethernet** datorită prețului relativ mic al interfeței LAN de acest tip. De asemenea, după realizarea fizică a interconectărilor prevăzute, va trebui începută înlocuirea software-elor de comandă existente. Aceste software-e, pe de o parte au la bază specificații închise, pe de altă parte nu au fost concepute pentru transmiterea comenzilor prin rețea LAN.

6.3.2. Rezultate obținute

Pe baza proiectului de modernizare amintit, au fost achiziționate până în prezent 5 interfețe LAN. La alegerea acestora s-a ținut cont de prețurile “destul” de ridicate ale interfețelor care utilizează metoda de transmisie a datelor de tip Jeton (Token Ring - IEEE802.4). În consecință s-a prevăzut achiziționarea de interfețe LAN care utilizează metoda CSMA/CD (“Ethernet” - IEEE802.3). Ele sunt de tip 10Base2 (“Thin Ethernet”) și au fost instalate în cele 5 calculatoare PC existente. Aceasta a permis interconectarea lor într-o rețea LAN de 10Mb/s, utilizând cablu coaxial.

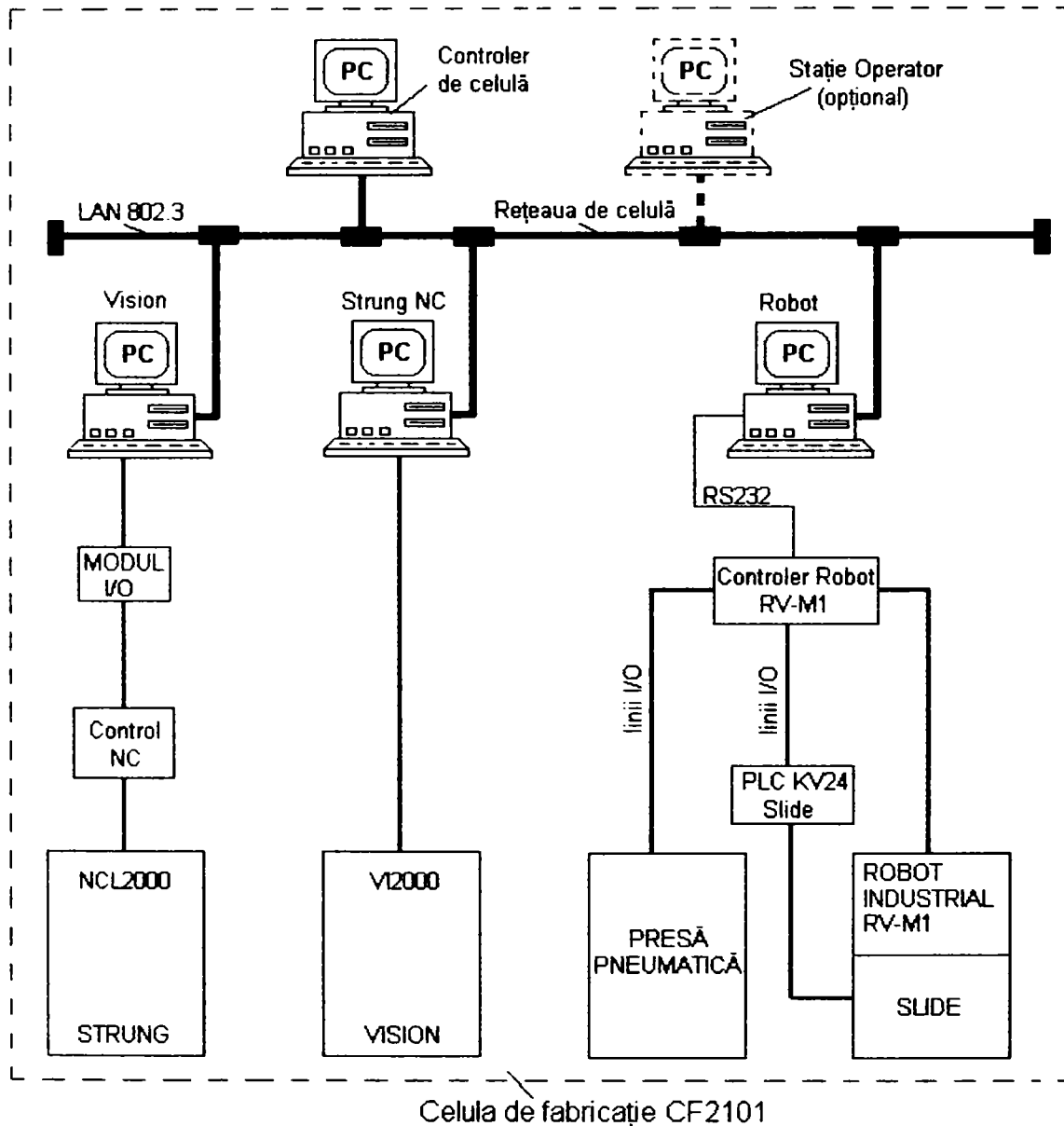


Fig.6.14. Rezultatul modernizării structurii de comandă a sistemului FMS2101, [12]

Cu aceste calculatoare PC, dotate deja cu interfețe LAN, cu o simplă reorganizare fizică și logică a fost posibilă implementarea (chiar și temporară) prevederilor de modernizare referitoare la o parte a mini-sistemului CIM2000: și anume la **modernizarea structurii de comandă a sistemului FMS2101**, care a devenit astfel “celula de fabricație CF2101”, (fig.6.14.). Pentru aceasta s-au utilizat doar 4 calculatoare, dar oricând există posibilitatea conectării la rețeaua de celulă din figură și al 5-lea calculator, care poate funcționa eventual pe post de “stație operator”.

Prin aceasta s-a creat posibilitatea începerii realizării unor structuri deschise de comandă, întâi la nivelul celulei CF2101 și apoi la nivelul întregului mini-sistem. În urma modernizărilor/adaptărilor efectuate asupra structurii de comandă, s-a constatat (cea ce era de fapt previzibilă) că, deși exista deja o interconectare fizică de tip LAN, între majoritatea stațiilor, totuși schimburile informaționale între aplicațiile originale au rămas inexistente. În consecință, pe lângă menținerea temporară și a vechilor interconectări informaționale, a devenit necesară **dezvoltarea treptată a unor noi structuri de transmitere a comenzilor** între stațiile mini-sistemului, de această dată de tip LAN (vechea structură utiliza pentru schimburile de informații între stații doar semnale discrete I/O). Până în prezent s-a reușit elaborarea:

- unor structuri de comandă de la distanță (adică prin controlerul de celulă) a robotului industrial RV-M1, fig.6.15,
- respectiv a unor structuri MMS, pentru întregul mini-sistem CIM2000.

Aceste structuri de comandă, elaborate pentru mini-sistemul CIM2000, sunt prezentate în paragrafele care urmează.

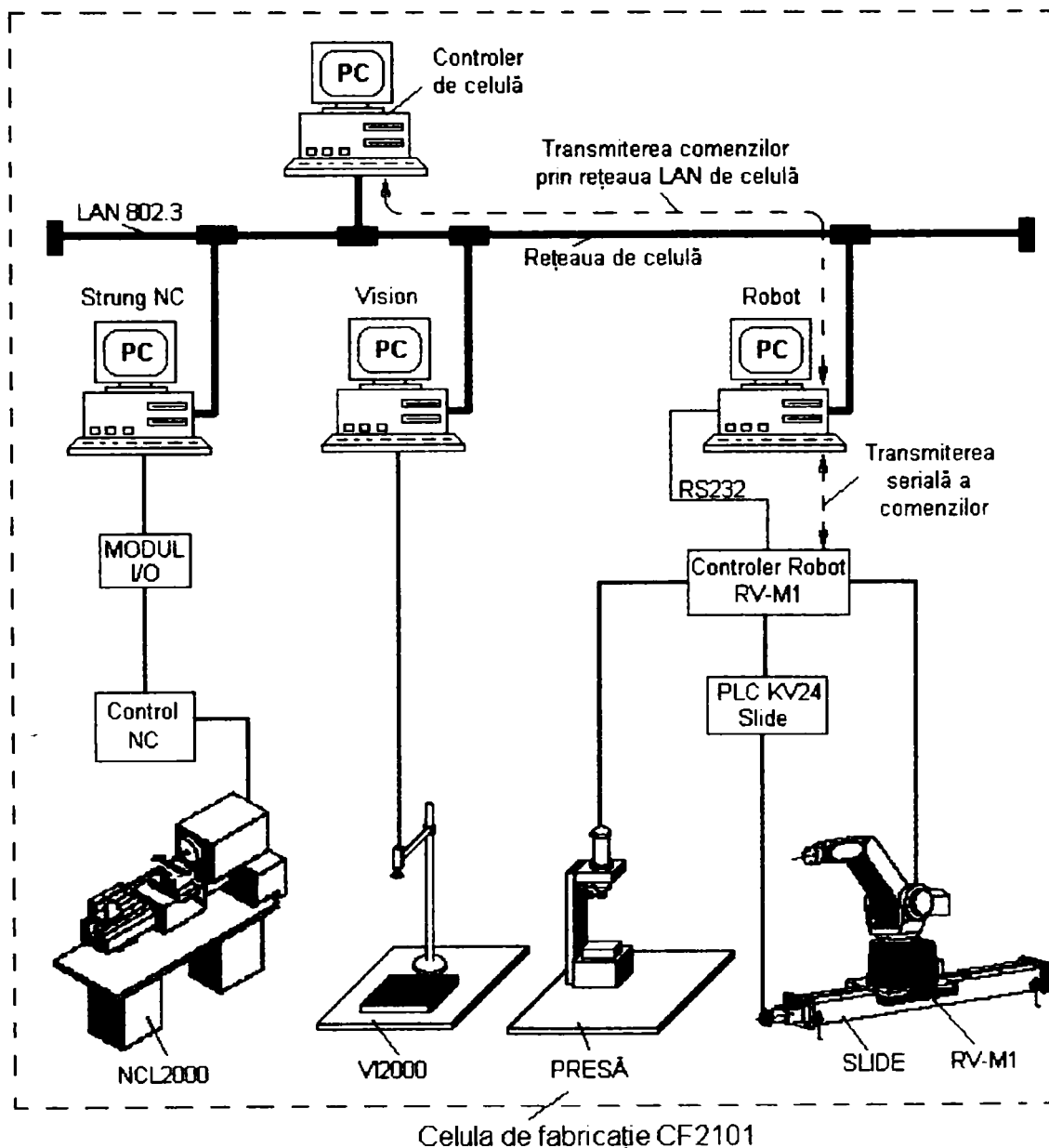


Fig.6.15. Transmiterea comenzilor de la controlerul de celulă la controlerul de robot RV-M1, [7]

6.4. Elaborarea unor structuri de comandă de la distanță a robotului RV-M1 din cadrul mini-sistemului CIM2000

La elaborarea unor structuri de comandă de la distanță a robotului RV-M1 era necesară în prima fază conectarea informațională a controlerului de robot la un calculator PC. Apoi se putea realiza conectarea acestui calculator "de interfață" la controlerul de celulă.

6.4.1. Conectarea informațională a controlerului de robot RV-M1 la un calculator PC

Conectarea controlerului de robot RV-M1 la un calculator PC s-a realizat, prin intermediul unei legături seriale, fig.6.16. Aceasta era posibilă deoarece atât calculatorul PC cât și controlerul de robot sunt dotate cu interfețe (seriale) RS232c.

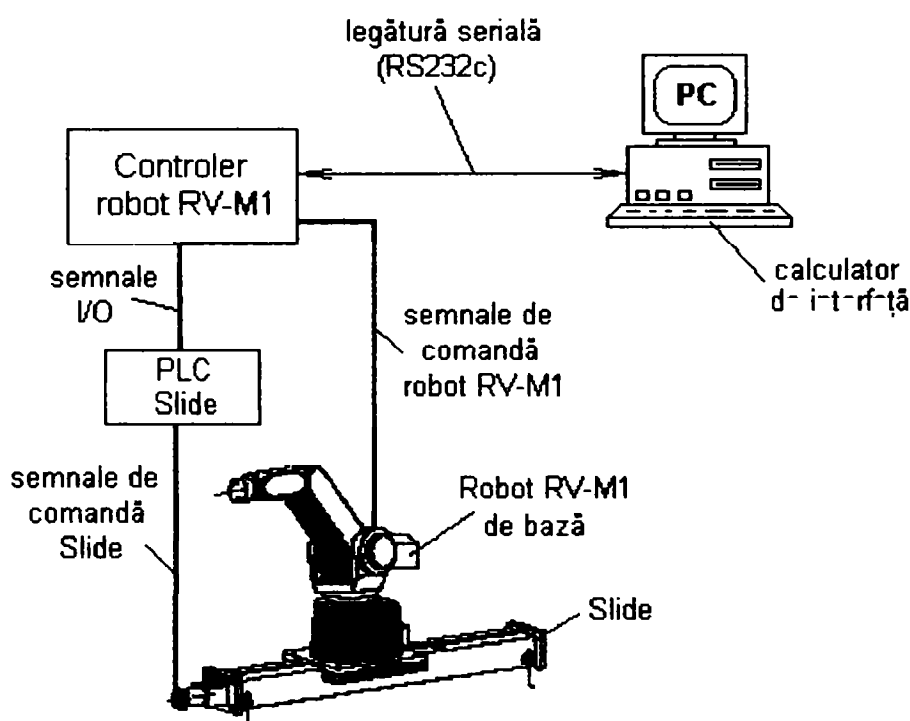


Fig.6.16. Conectarea controlerului de robot RV-M1 la un calculator PC, [8]

Pe lângă conectarea fizică (efectuată prin intermediul unui cablu de legătură serială), era necesară realizarea unui program de transmitere serială a comenzilor (specifice robotului RV-M1), de la calculatorul PC la controlerul de robot. Principalele comenzi ale robotului RV-M1, care pot fi transmise de la un calculator, sunt trecute în tabelul 6.1.:

Comanda	Sintaxa comenzii	Semnificație
NT (Nest)	NT	Sincronizare robot.
MO (Move)	MO <nr.poziție>, [<stare gripper>]	Deplasare (în regim punct cu punct) în poziția specificată prin număr.
MS (Move Straight)	MS <nr. poziție, nr. puncte intermediare>	Deplasare în regim multipunct în poziția indicată prin număr.
MP (Move Position)	MP <X, Y, Z, P, R>	Deplasarea într-o poziție indicată prin coordonatele carteziene absolute.

DW (Draw)	DW <dX, dY, dZ>	Deplasarea într-o poziție indicată prin coordonatele relative față de poziția curentă.
MJ (Move Joint)	MJ <J1, J2, J3, J4, J5>	Determină mișcarea fiecărei articulații J față de poziția curentă.
MT (Move Tool)	MT <nr. poziție, mărime deplasare, stare gripper>	Determină deplasarea sculei (cu valoarea specificată) de-a lungul axei acesteia.
GO (Gripper Open)	GO	Deschiderea dispozitivului de prehensiune.
GC (Gripper Close)	GC	Închiderea dispozitivului de prehensiune.
HE (Here)	HE <nr. poziție>	Determină memorarea coordonatelor carteziene absolute ale poziției curente.
PD (Position Define)	PD <nr.poz., X, Y, Z, P, R>	Definirea unei poziții, indicând valorile coordonatelor carteziene absolute.
WH (Where)	WH	Citirea din memorie a coordonatelor carteziene ale poziției curente.
PR(Position read)	PR <nr. poziție>	Citirea din memorie a coordonatelor carteziene ale unei poziții.
LR(Line read)	LR <nr. frază>	Citirea din memorie a unei fraze de program indicat prin număr.
RN (Run)	RN <nr. frază>	Lansarea programului de aplicație din memorie, indicând fraza de început.

Tab.6.1. Principalele comenzi ale robotului RV-M1, [85]

Pe lângă partea de "comunicație", trebuia realizată și o "interfață cu utilizatorul". Aceasta avea rolul preluării comenzilor de la un operator și transmiterea acestora către controlerul de robot. Programul a fost realizat sub sistemul de operare DOS, în limbaj Pascal. Pentru interfață cu utilizatorul s-au utilizat funcții predefinite din Turbo Vision. Astfel meniurile și submeniurile realizate sunt asemănătoare mediului de lucru Turbo Pascal.

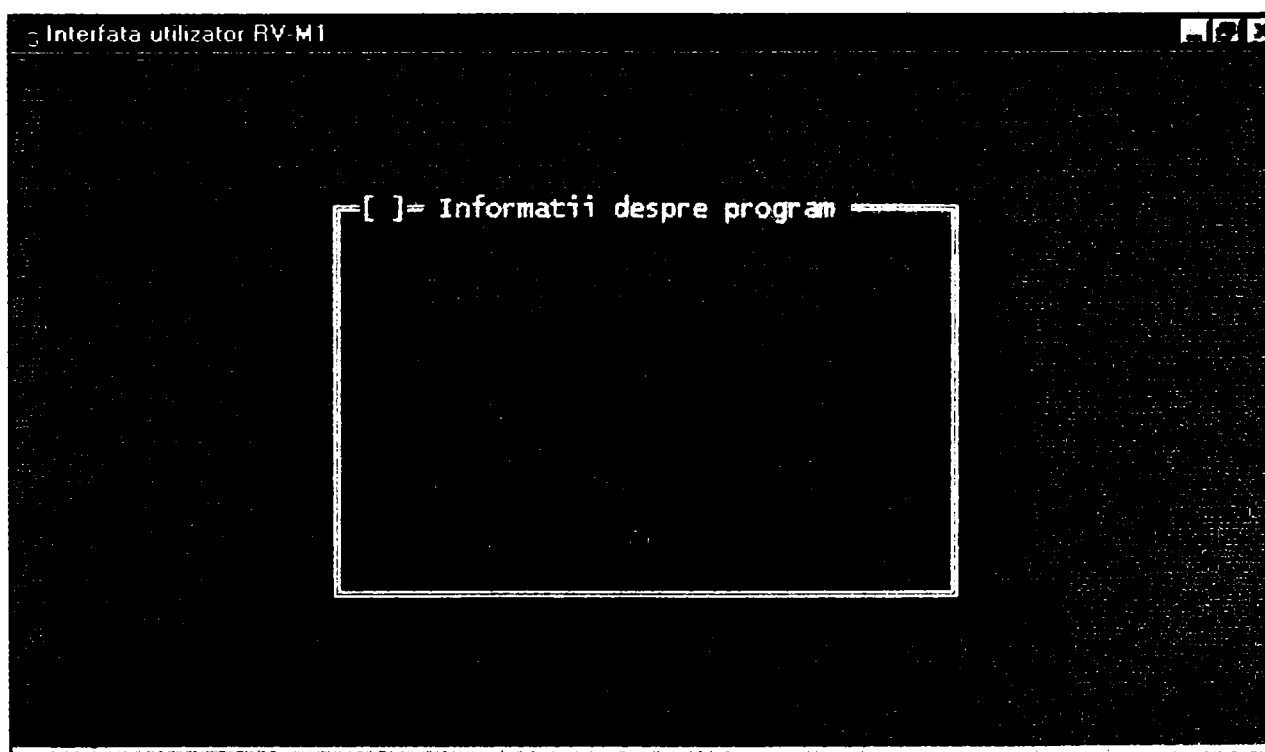


Fig.6.17. Interfața utilizator a programului de comunicație

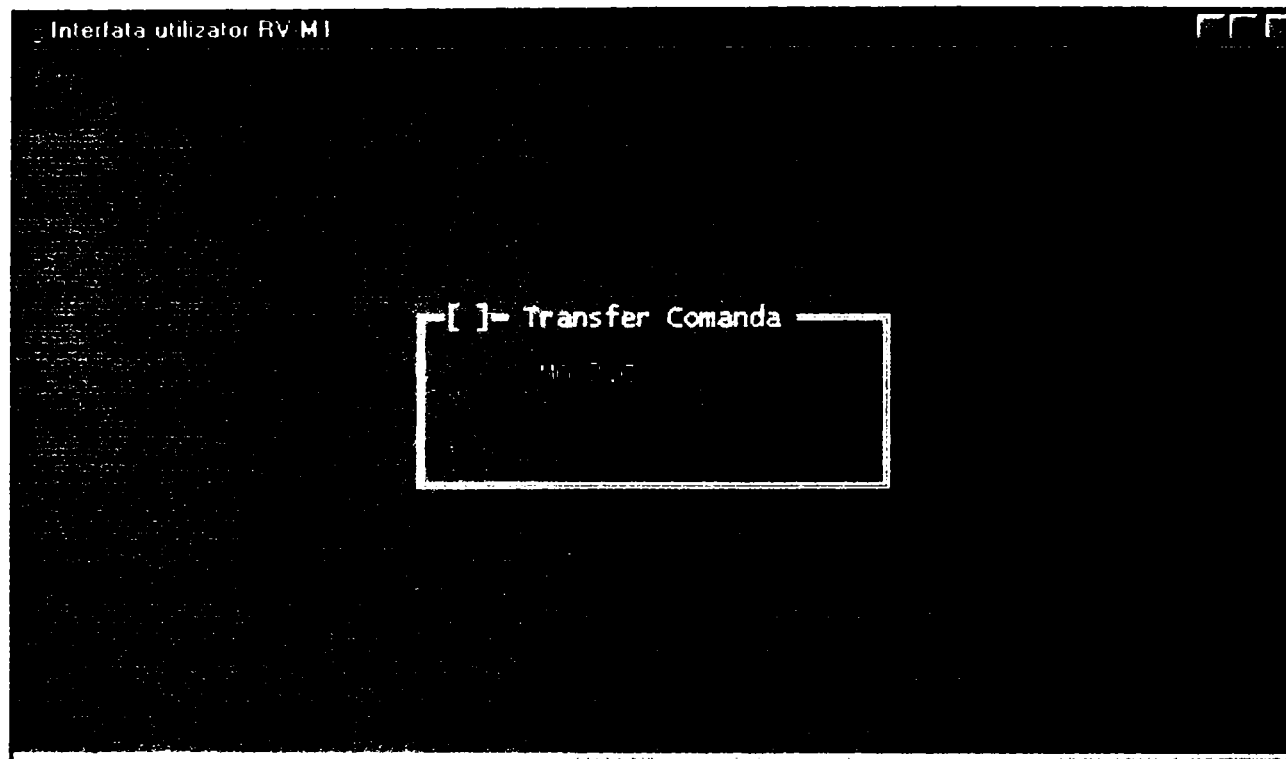


Fig.6.18. Fereastra de transmitere comenzi



Fig.6.19. Submeniul de transfer programe

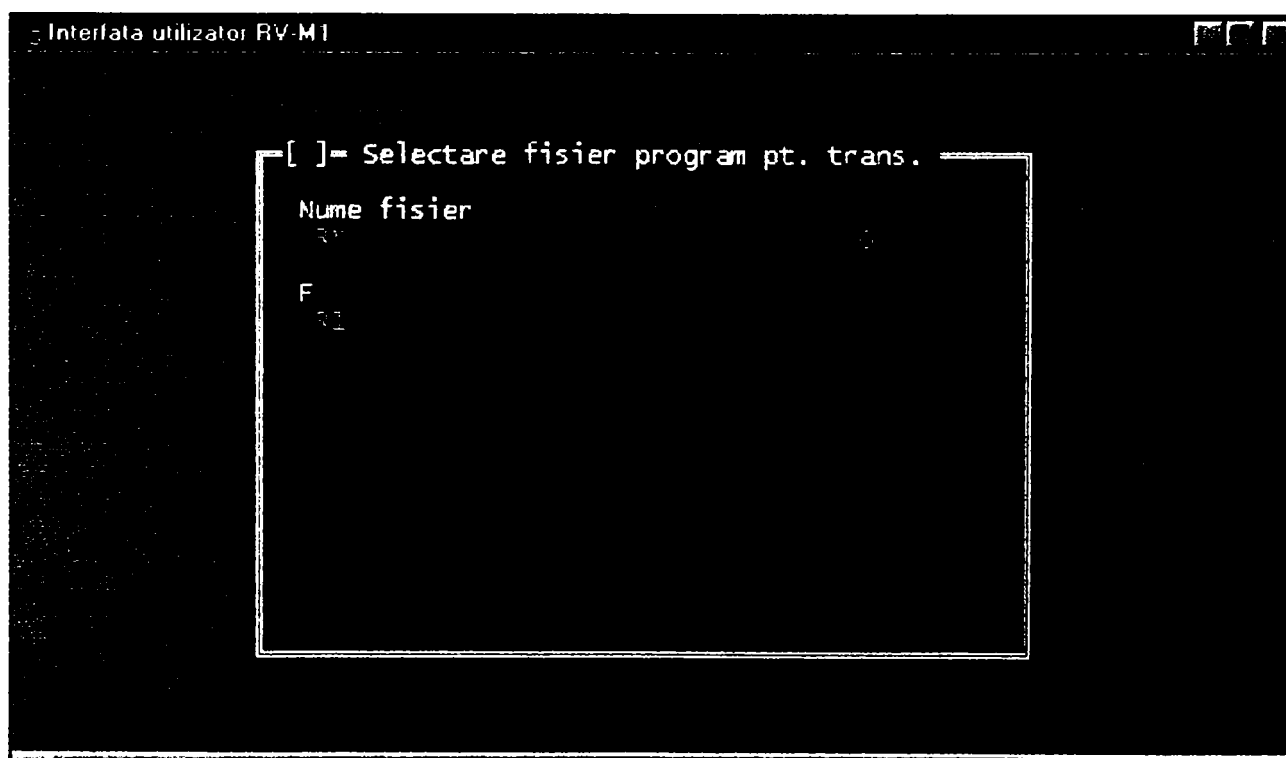


Fig.6.20. Fereastra de selectare program de aplicație pentru transfer

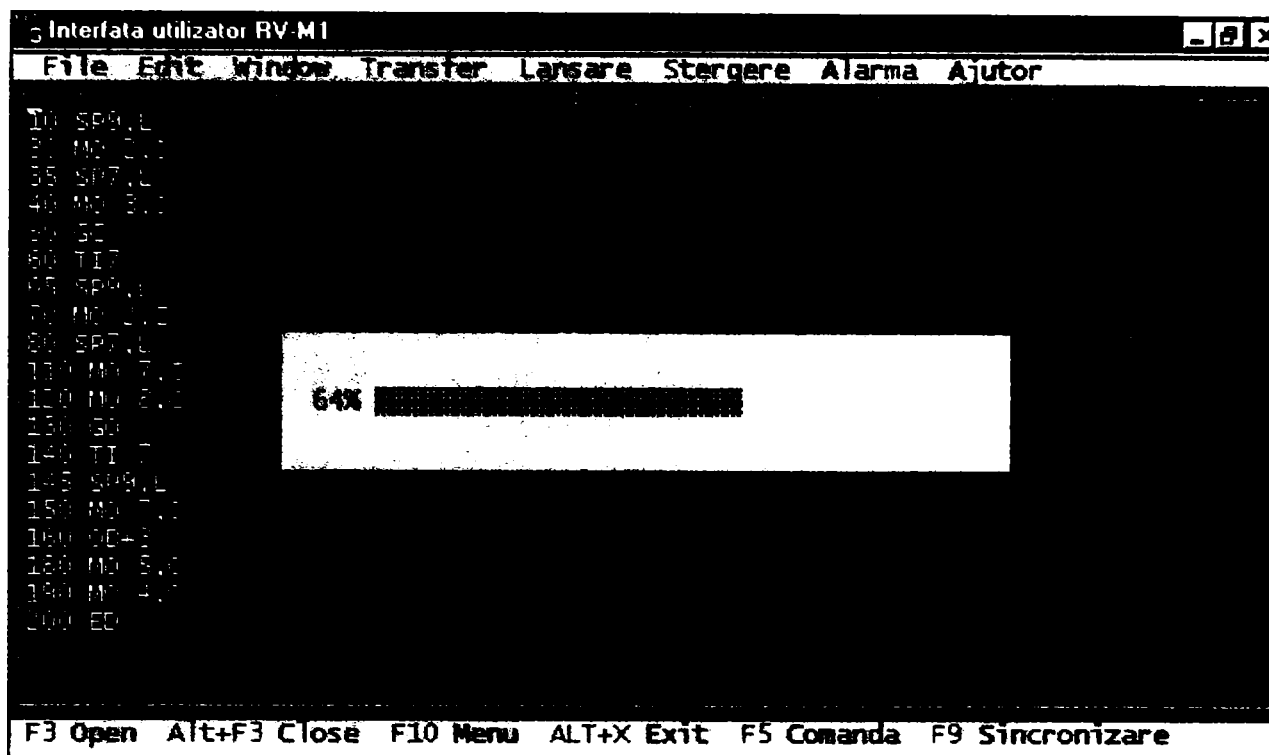


Fig.6.21. Fereastra de transfer program de aplicație

Programul realizat, pe lângă *transferul comenzilor* (fig.6.18.), permite și *transferul programelor de aplicație* de la PC către robot (și invers), fig.6.19. În acest scop o fereastră dedicată ajută la selectarea fișierului corespunzător, fig.6.20. Odată cu realizarea transferului, fișierul selectat este deschis pentru vizualizare într-un editor de texte încorporat, fig.6.21. Acest editor poate fi utilizat și la crearea/editarea programelor de aplicație. Alte funcții ale programului:

1. **operații de transfer** fișier cu coordonatele pozițiilor memorate/definite;
2. **operații de lansare** programe de aplicație existente în memoria controlerului robot;
3. **operații de ștergere** programe și coordonate de poziții din memoria controler-ului robot;
4. **operații de anulare semnalizare alarmă** în caz de erori de sintaxă;
5. **sincronizarea robotului**, readucerea articulațiilor în originea lor mecanică.

Observații:

1. Transferul unei comenzi se realizează prin **transmiterea succesivă a caracterelor ASCII** din cadrul comenzii, [85]. De exemplu în cazul comenzii "MO 2,C" (deplasare punct caracteristic în poziția nr.2 cu dispozitiv de prehensiune închis - Close), se transmit pe rând caracterele "M", "O", "2", ",", și "C".
2. Dacă comanda care se transmite este precedată de un număr de ordine, de exemplu "30 MO 2, C" atunci ea este considerată de către controlerul robotului frază de program. În consecință după recepție nu este executată imediat ci este înmagazinată în memoria controlerului de robot sub forma unei linii de program.
3. Transmiterea de la calculatorul PC a unui fișier cu program de aplicație, se realizează de fapt prin transmiterea succesivă a frazelor programului respectiv.
4. Pentru "**salvarea**" din memoria controlerului de robot a unui program de aplicație, se utilizează comanda LR (Line Read), prin care se citesc pe rând toate frazele programului, [85].

6.4.2. Elaborarea unor structuri de comandă pentru accesarea de la distanță a resurselor fizice și logice ale robotului RV-M1

În capitolul 5 s-a elaborat un model de accesare de la distanță a resurselor fizice și logice ale unui robot industrial. Trebuie arătat că pentru implementarea acestuia ar fi fost necesar un controler de robot cu arhitectură deschisă, care să permită adaptarea structurii de comandă în concordanță cu modelul respectiv. Ne având la dispoziție un astfel de controler, s-a încercat implementarea parțială a unor structuri din cadrul acestui model, pe calculatorul de interfață a robotului RV-M1. Astfel, între calculatorul PC și controlerul de robot a rămas protocolul vechi de comandă, stabilit de firma producătoare Mitsubishi, însă legătura cu celelalte structuri de comandă din cadrul mini-sistemului CIM (cel puțin parțial) respectă acest model.

Pentru implementare s-a creat o **structură de comandă cu arhitectură client-server**, în care server-ul este sistemul robot RV-M1 (format din robotul propriu-zis, controlerul de robot și calculatorul de interfață), iar client-ul este un alt calculator PC, pe rol de controler de celulă, fig.6.22.

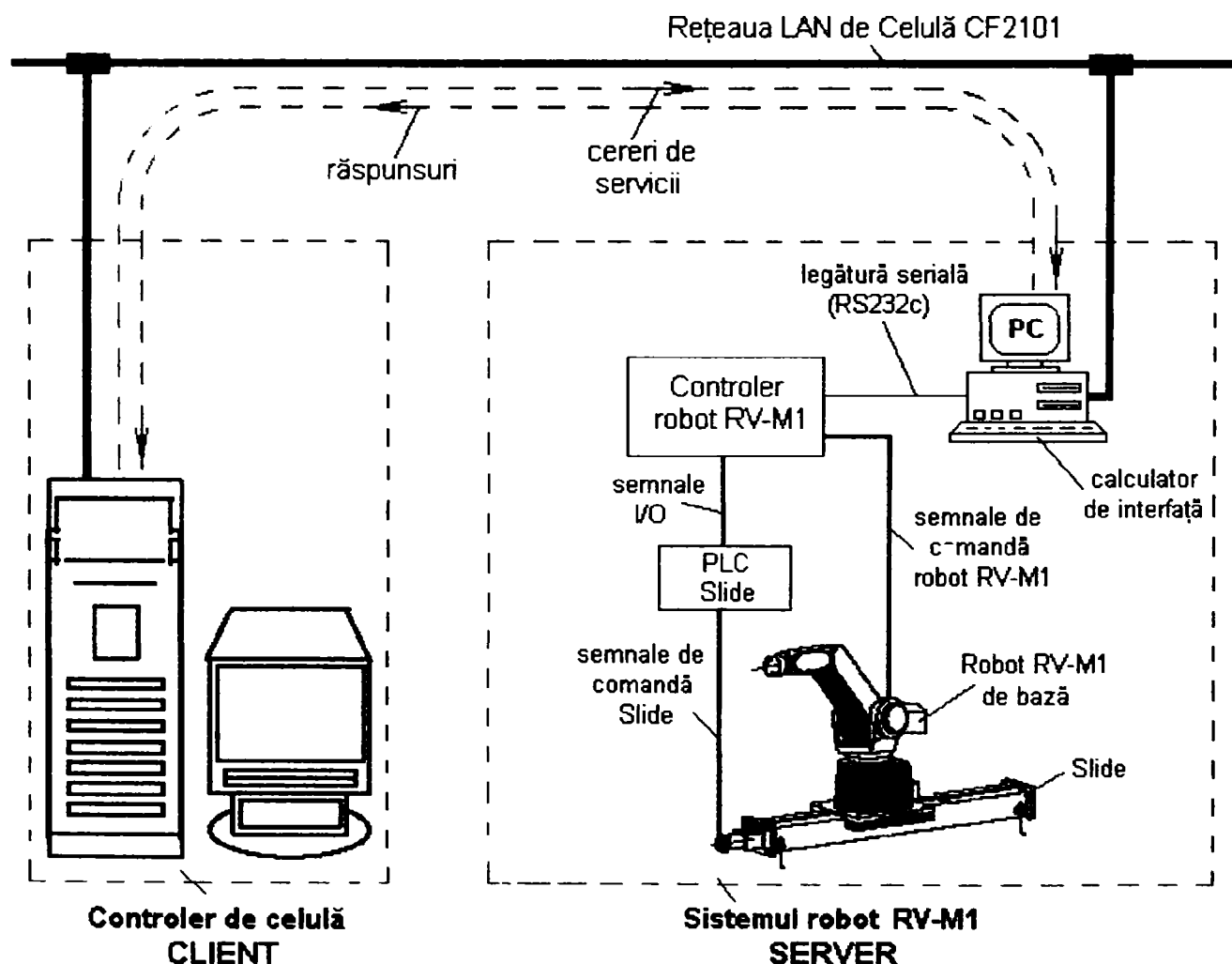


Fig.6.22. Structura de comandă cu arhitectură client-server, [8]

Pentru legătura între cele două calculatoare PC este utilizată rețeaua LAN de tip Ethernet din cadrul celulei CF2101. Prin această rețea sunt transmise comenzile sub forma unor cereri de servicii din partea controlerului de celulă și răspunsuri din

partea sistemului robot, fig.6.22. În tabelul 6.2. s-a reprezentat lista cererilor de servicii implementate.

Observație:

- Trebuie arătat că aceste cereri de servicii sunt apelul serviciilor prevăzute în modelul de accesare de la distanță a resurselor fizice și logice ale roboților industriali, elaborat și prezentat în capitolul 5. Ele au fost concepute astfel încât să ofere un mod standard de comandă de la distanță, a oricărui robot industrial, indiferent de producător. Astfel ele stau la baza realizării unor structuri deschise de comandă în Hipersisteme CIM.

Cererea de serviciu	Sintaxa stabilită	Semnificație
LOAD	Load, <nume program>	Încărcare date/program de aplicație în memoria controlerului robot
CLEAR	Clear, <nume program>	Ștergere date/program de aplicație din memoria controlerului robot
SELECT	Select, <nume program>	Selectare program de aplicație în vederea lansării în execuție
START	Start	Lansare în execuție program de aplicație
RESUME	Resume	Relansarea în execuție a programului de aplicație terminat.
STATUS	Status	Citire/preluare stare de funcționare sistem robot
GET ATTRIBUTES	GetAttributes, <nume modul>	Obținere listă cu atributele ale unui modul definit.
READ	Read, <nume modul, nume atribut>	Citirea valorii unui atribut al unui modul specificat.
WRITE	Write, <nume modul, nume atribut, valoare>	Modificarea valorii unui atribut indicat prin nume.

Tab.6.2. Lista cererilor de servicii care au fost implementate

Cererile de servicii implementate servesc pe de o parte la **operații cu programe de aplicație** (încărcare de pe controler de celulă, selectare, lansare în execuție etc.) pe de altă parte servesc la **accesarea atributelor modulelor**. Astfel, cu cererea de serviciu GET ATTRIBUTES, controlerul de celulă poate să obțină o listă cu denumirile atributelor unui modul, iar cu serviciile READ și WRITE poate accesa (prin citire sau scriere) valoarea unui atribut.

Controlerul de celulă, utilizând astfel un limbaj standard al "cererilor de servicii", numai utilizează pentru comanda (de la distanță) a robotului RV-M1 comenzile definite de producător (adică pe cele din tabelul 6.1.). Pentru punerea în mișcare a robotului utilizează cererile de servicii indicate în tabelul 6.2, prin care realizează încărcarea și lansarea în execuție a unui program adecvat de aplicație.

Accesarea atributelor modulelor definite are rol în monitorizarea stărilor de funcționare și luarea deciziilor în funcție de valoarea actuală a atributelor respective. Astfel, controlerul de celulă poate afla coordonatele situației curente, prin citirea valorii atributului "Situarea Curentă Sculă", al modulului Controler Dispozitiv de Ghidare.

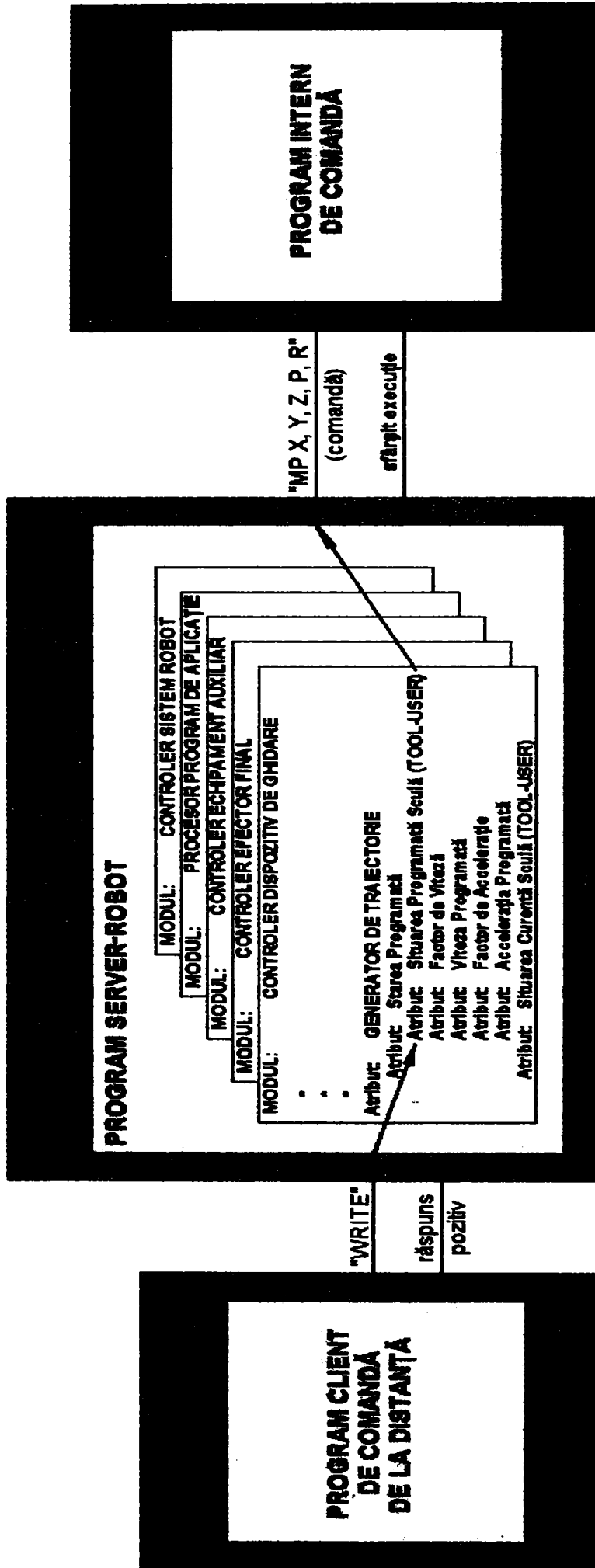


Fig.6.23. Principiul comenzii robotului RV-M1 prin accesarea atributelor modulelor definite

Prin modificarea valorii atributului "Situarea Programată Sculă" al aceluiași modul, controlerul de celulă poate să prescrie o mișcare a dispozitivului de ghidare, în situația indicată prin aceasta valoare. În cazul în care mișcarea este permisă (adică atributul "Permis Mișcare" are valoarea ADEVĂRAT), calculatorul de interfață traduce această cerere de serviciu în limbajul propriu al robotului RV-M1, și emite către controlerul de robot o comandă de mișcare de tip *MP (Move Position)* indicând coordonatele X, Y, Z, P și R ale situației țintă, fig.6.23.

Pentru accesarea resurselor robotului RV-M1, prin "comenzi" transmise de la controlerul de celulă, s-a realizat două programe. *Programul Client* este rulat pe controlerul de celulă și preia comenzile de la un operator, le transformă în cereri de servicii și le emite (prin rețeaua LAN), către celălalt program numit *program Server-Robot*. Acesta din urmă este rulat pe calculatorul de interfață robot, recepționează cererea de serviciu, execută comanda aferentă și transmite înapoi către programul Client un răspuns pozitiv (în cazul în care cererea de serviciu s-a putut executa) sau a unui răspuns negativ (în cazul în care cererea nu s-a putut executa).

Programul Server-Robot s-a obținut prin completarea programului de comunicație serială prezentat în paragraful anterior cu structurile de date aferente modulelor definite și cu funcțiile aferente transiterii și recepționării "comenzilor" prin rețea LAN, fig.6.24.

S-a redimensionat fereastra editorului de texte și s-a definit în plus încă o fereastră care indică un set de informații legat de stările de funcționare ale sistemului robot, fig.6.25. Astfel utilizatorul este informat despre existența legăturilor informaționale cu controlerul de robot respectiv cu controlerul de celulă (ONLINE/SERVER), despre starea sincronizată a robotului, despre numele programelor încărcate, selectate, rulate etc. De asemenea este vizualizată starea de funcționare aferentă echipamentului de comandă robot ("LOADED" – adică ÎNCĂRCAT cu programul R3).

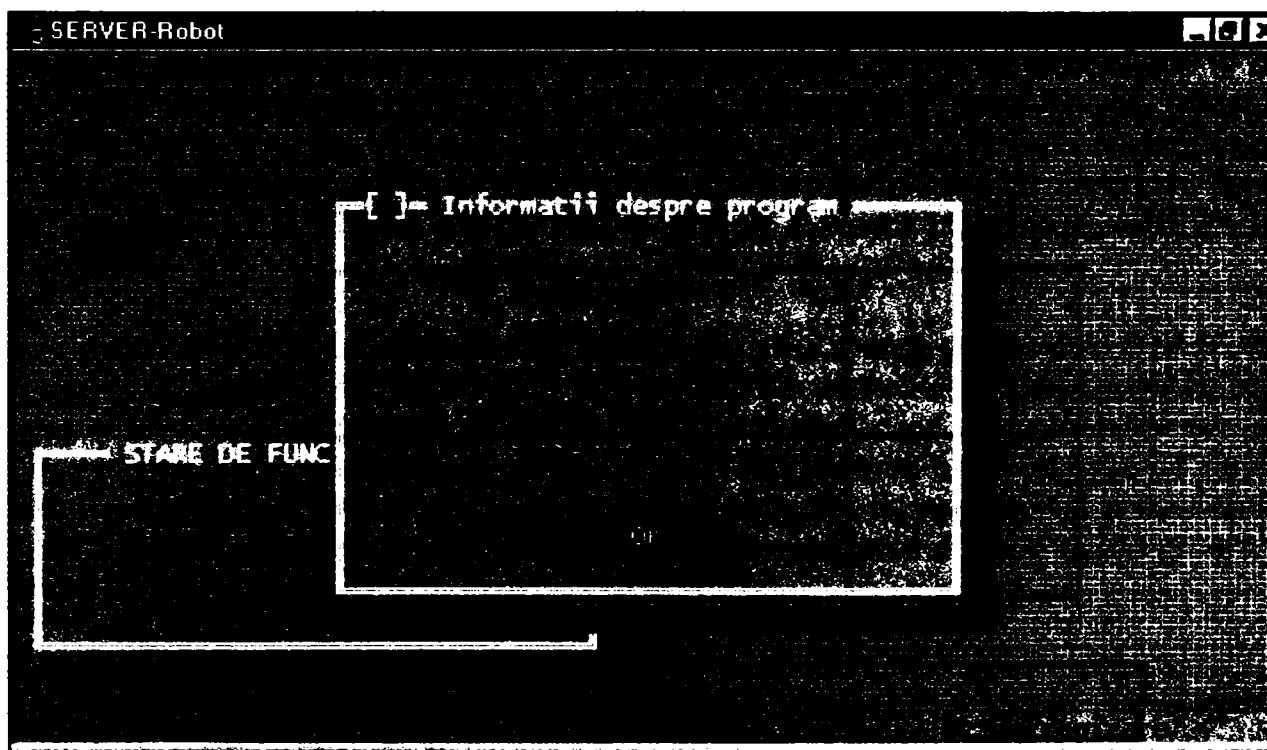


Fig.6.24. Interfața utilizator a programului Server-Robot

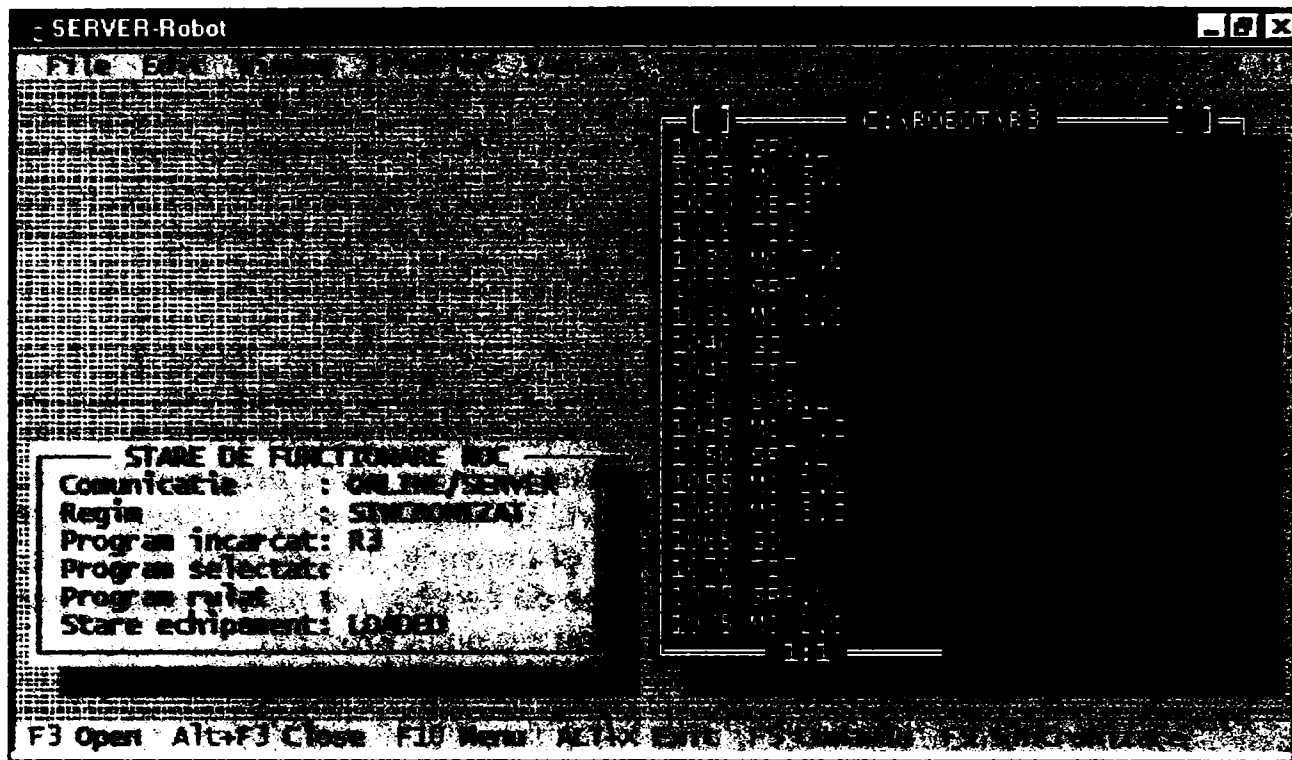


Fig.6.25. Fereastra cu stările de funcționare și fereastra editorului de texte

Pentru transmiterea prin rețeaua Ethernet a mesajelor corespunzătoare cererilor de servicii, precum și a răspunsurilor aferente, s-a încercat utilizarea unei *arhitecturi deschise*, oferită de Specificația Driver-ului de Pachete (Packet Driver Specification), [39]. Acesta oferă pentru orice utilizator un mod de acces deschis la pachetele de date transmise printr-o rețea LAN.

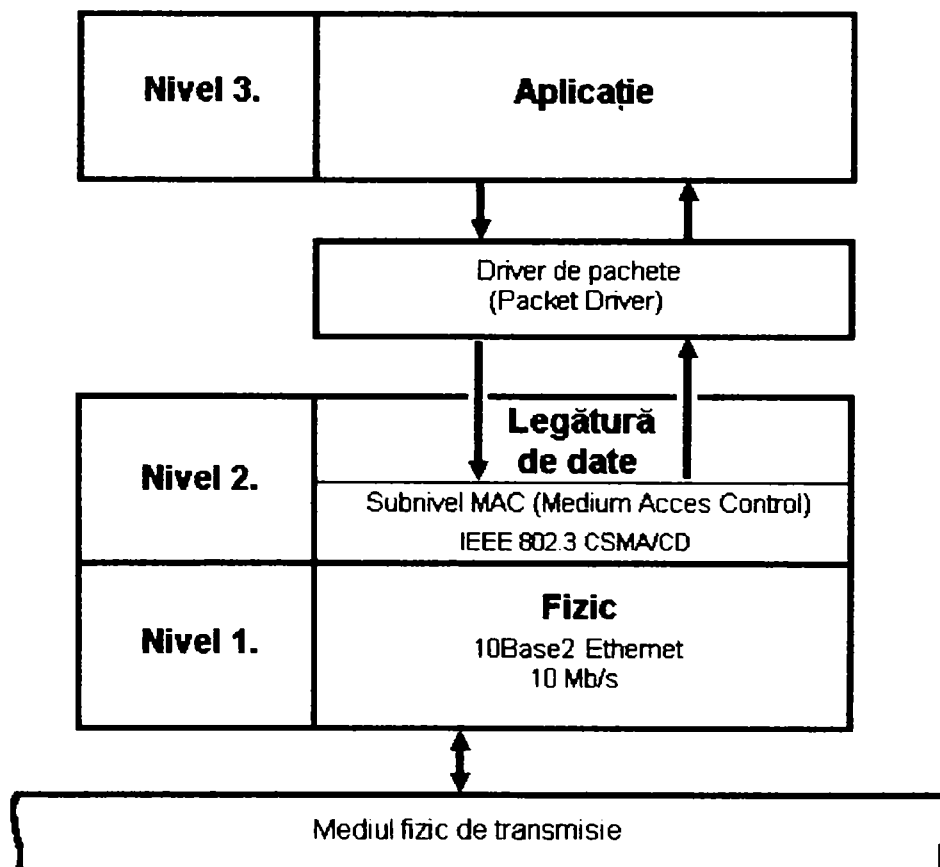


Fig.6.26. Arhitectura utilizată la transmiterea mesajelor prin rețeaua LAN

Arhitectura utilizată este organizată pe 3 nivele, în care “aplicația” program (de exemplu programul Client al controlerului de celulă) transmite pachete de date către o altă aplicație program (de ex. către programul Server-Robot), accesând mediul de comunicație prin “driver-ul de pachete”. Acesta din urmă, în general este comercializat împreună cu “celelalte” driver-e ale interfeței de rețea LAN. Accesul la rețea se realizează la nivelul legăturii de date, subnivelul **MAC (Medium Access Control)**, fig.6.26. La acest nivel formatul pachetelor de date transmise este următorul:

Adresa stației destinație [6 octeți]	Adresa stației sursă [6 octeți]	Lungime (tip) pachet [2 octeți]	Datele aferente mesajului transmis [46... 1500 octeți]
--	---------------------------------------	---------------------------------------	--

Fig.6.27. Formatul pachetului de date la nivel MAC

Pe baza figurii, mărimea maximă a unui mesaj care se poate transmite este de 1500 de octeți (1500 de caractere). La transmiterea unui mesaj de la controlerul de celulă la calculatorul de interfață robot, programul Client întâi formează un pachet de date, în care mesajul propriu-zis este precedat de adresa Ethernet a stației destinație (de exemplu “00:00:21:60:80:46” care este de fapt adresa plăcii de rețea Ethernet din calculatorul de interfață robot), de adresa stației sursă (de exemplu: “00:C0:DF:C2:88:D6” care este de fapt adresa plăcii de rețea din controlerul de celulă) și de tipul pachetelor transmise (\$0800 pentru Ethernet). Apoi apelează funcția standardă de transmitere (send) a pachetului format.

Trebuie menționat că aceeași structură a pachetului de date este utilizată și la transmiterea inversă a informațiilor, adică de la calculatorul de interfață la controlerul de celulă.

6.4.3. Implementarea diagramei stărilor de funcționare

Prin intermediul programului Client realizat se poate transmite deci de la controlerul de celulă cererile de servicii prezentate în tabelul 6.2. După fiecare cerere recepționată și executată, programul Server-Robot răspunde cu o listă aferentă stărilor modificate de funcționare. Controlerul de celulă astfel este informat nu numai despre execuția serviciului solicitat dar și despre **starea curentă de funcționare a sistemului robot**.

Controlerul de celulă la nevoie poate utiliza și cererea de serviciu STATE (stare) pentru aflarea stărilor curente de funcționare ale sistemului robot. Aceasta poate fi utilă atunci când controlerul de celulă (într-un eventual mod de lucru automat), este în așteptarea unui eveniment care are ca efect modificarea stării curente de funcționare. De exemplu a fost lansat în execuție un program de aplicație și controlerul de celulă așteaptă terminarea acestuia, interogând la perioade de timp egale programul Server-Robot.

Observație:

- În timpul rulării unui program de aplicație, sistemul robot se află în starea de ROBOT-ÎN-EXECUȚIE (ROBOT- EXECUTING), iar după terminarea programului intră în starea READY.

Pentru a se putea realiza "monitorizarea" de la distanță a stărilor de funcționare ale sistemului robot, s-a încercat **implementarea diagramei stărilor de funcționare** prezentat în capitolul 5., fig.5.9. La implementare însă au apărut două probleme:

1. Controlerul de robot RV-M1 permite încărcarea în memorie numai a unui singur program de aplicație. Nefiind astfel necesară selectarea programului, robotul după încărcare intră direct într-o stare (corespunzătoare stării READY), în care se poate lansa în execuție singurul program din memorie. Astfel, în cazul controlerului robot RV-M1, stările ROBOT-ÎNCĂRCAT și ROBOT-READY sunt suprapuse.
2. Producătorul nu a definit nici o comandă externă (transmisă prin legătură serială), prin care să se poate suspenda temporar execuția programului de aplicație. Astfel robotul nu poate intra niciodată în starea de "PAUZĂ-MIȘCARE-ROBOT".

Prima problemă s-a putut rezolva prin "**segmentarea**" memoriei aferentă **programelor de aplicație** (16KB) astfel încât să permită încărcarea a mai multor programe mai "scurte". Aceasta era posibilă printr-un artificiu "extern", adică fără vreo intervenție în arhitectura hardware/software a controlerului. Ideea care a condus la rezolvarea problemei are la bază modul de gestiune a frazelor de program de către controlerul de robot RV-M1.

Astfel, în cazul unui program de aplicație, fiecare frază este numerotată, de exemplu: "30 MO 3, O" (frază numărul 30, care conține o comandă de deplasare în poziția 3 cu dispozitiv de prehensiune deschis – Open). În memorie poate exista fraze cu numere de ordine între 1 și 2048.

Pentru lansarea în execuție a programului trebuie indicat, pe lângă comanda aferentă (RN – Run) și numărul frazei de început, de exemplu "RN 10". După lansare sunt executate pe rând toate frazele din memorie, începând cu fraza de start (în cazul de față fraza 10). Bineînțeles un program se poate lansa de la oricare altă frază, chiar și de la fraza 30.

Execuția are loc până când controlerul programului de aplicație găsește **prima frază cu instrucțiunea ED** (End) care semnifică sfârșit program, de exemplu "420 ED". Trebuie arătat că în memorie pot exista și alte fraze, cu număr de ordine mai mare decât 420, dar ele nu sunt executate, fig.6.28. Acuma se vede deja, că practic prin utilizarea a mai multor instrucțiuni ED pot fi delimitate "programele" din memorie. De exemplu un program poate utiliza domeniul "de adrese" între 1-500, al doilea program între 501-1000, al treilea program

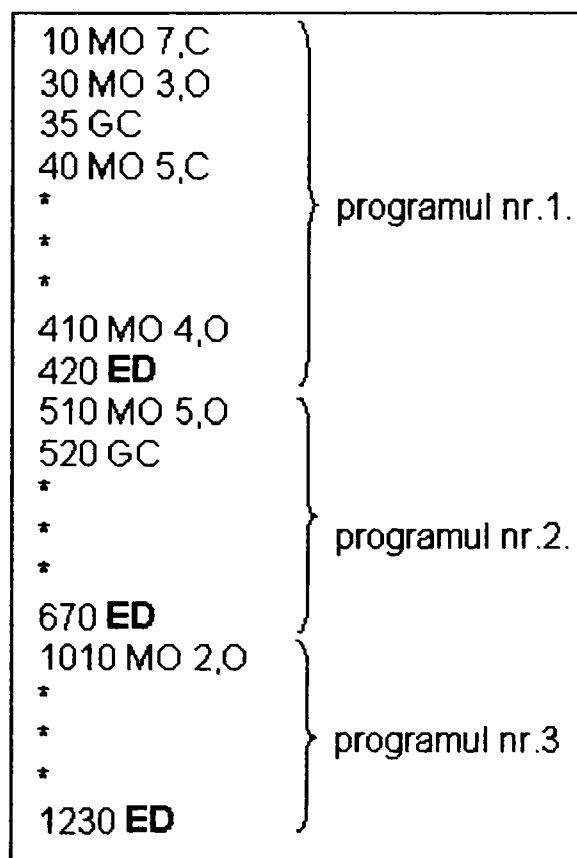


Fig.6.28.

1001-1500 etc. O condiție necesară este ca fiecare program să se termine cu o instrucțiune ED, fără depășirea domeniului alocat.

Cu mențiunile de mai sus s-a reușit implementarea diagramei stărilor de funcționare în următoarea formă:

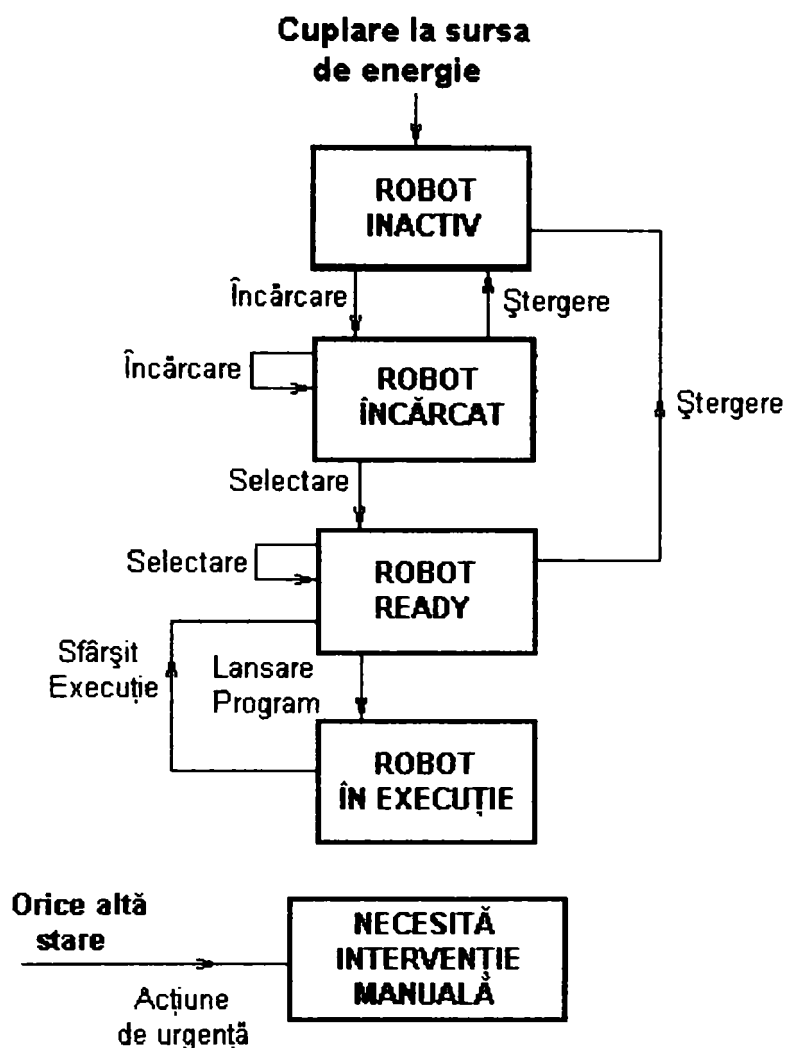


Fig.6.29. Diagrama stărilor de funcționare implementată pentru robotul RV-M1

Conform figurii 6.29., controlerul permite deci încărcarea a mai multor programe de aplicație, însă prin serviciul de ștergere (clear) sunt eliminate simultan toate programele din memorie. Astfel prin acest serviciu, **din starea de ROBOT-READY intră direct în starea ROBOT-INACTIV**. De asemenea se poate observa că nu a fost implementată și în consecință lipsește starea de “PAUZĂ-MIȘCARE-ROBOT”. O tranziție de intrare în aceasta stare se poate iniția doar printr-o comandă locală, respectiv manuală. Figura 6.29. însă conține numai acele stări, care pot fi rezultatul unor comenzi date “de la distanță”.

După implementarea acestor stări de funcționare în programul Server-Robot, **a fost completat programul Client** cu o funcție denumită “Robot operation state” prin care se putea afla și afișa starea curentă de funcționare a sistemului robot. Prin activarea funcției respective de către un operator, s-a inițiat practic transmiterea unei cereri de serviciu STATE (stare) către calculatorul de interfață robot. În fig.6.30. este prezentat momentul când este afișată, într-o fereastră dedicată, starea curentă de funcționare a sistemului robot.

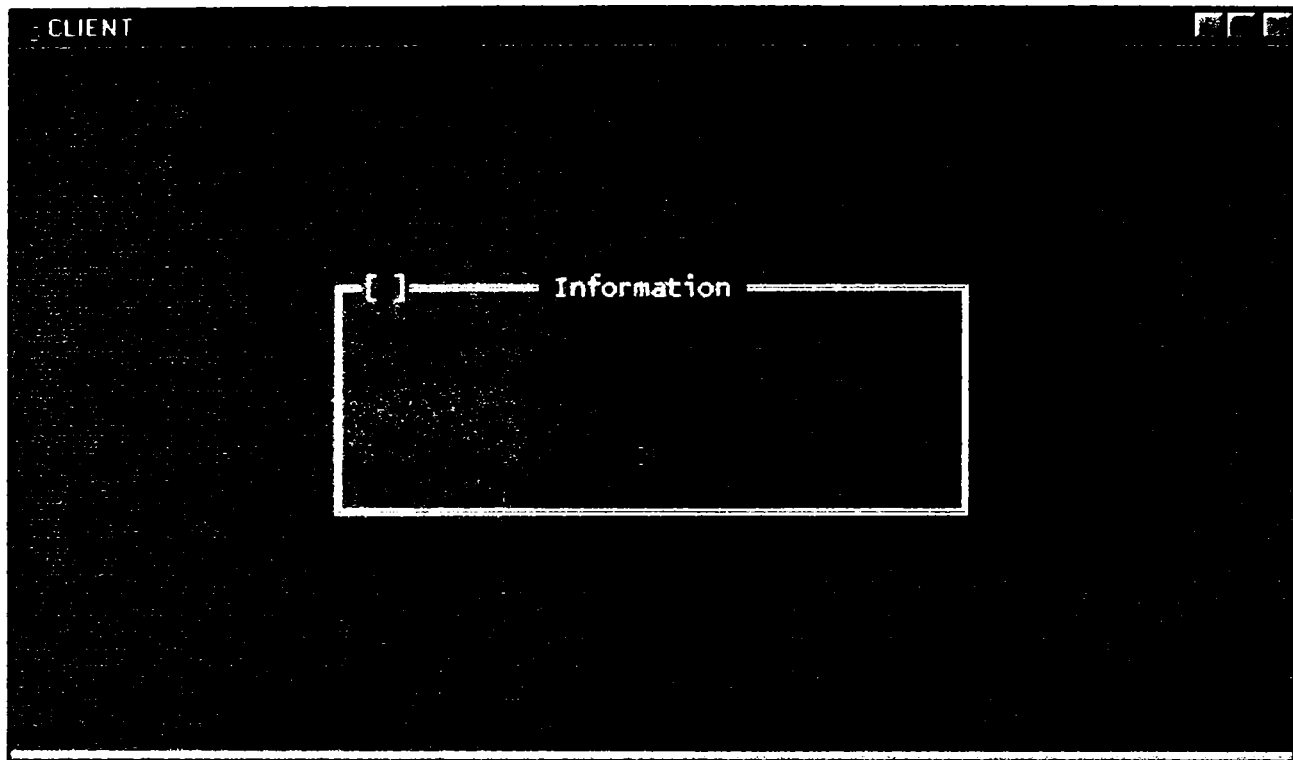


Fig.6.30. Interfața programului Client completat cu meniul "Robot operation state", [9]

Cu acest program Client, de asemenea se pot lansa și celelalte comenzi (prin meniul Commands), aferente cererilor de servicii din tabelul 6.2.

6.5. Elaborarea unor structuri deschise MMS de semaforizare a accesului la resurse comune în cadrul mini-sistemului CIM2000

Semaforul este un instrument al MMS-ului, cu ajutorul căruia se poate împărți între solicitanți numărul limitat de resurse fizice și logice existente, ele devenind astfel resurse comune.

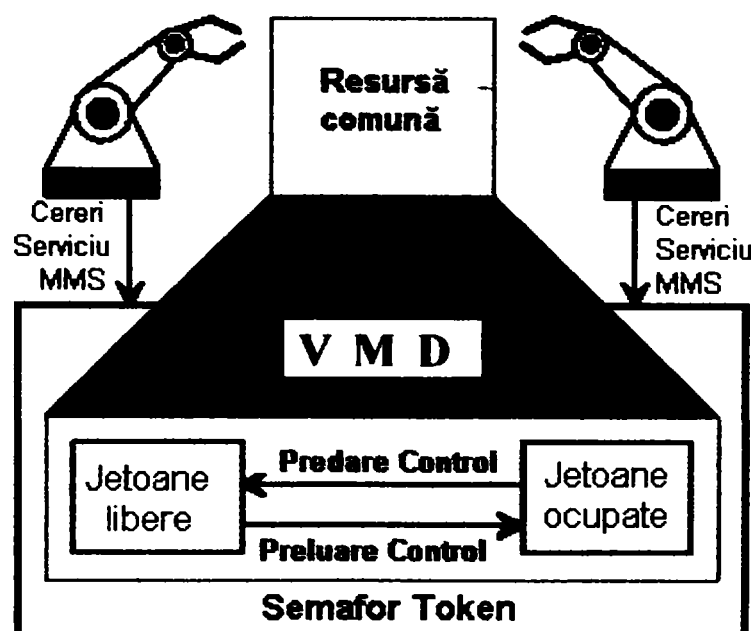


Fig.6.31. Principiul utilizării semaforului de tip Token, [102, 103]

La fiecare resursă se asociază câte un semafor, astfel încât accesul unui utilizator la resursa respectivă este condiționată de preluarea controlului asupra semaforului corespunzător.

În funcție de natura lor, resursele pot fi utilizate simultan de unu sau mai mulți utilizatori. Pentru a facilita acest lucru, la fiecare resursă se asociază, în cadrul semaforului, un număr de *jetioane* egal cu numărul maxim de utilizatori care pot avea acces simultan la resursa respectivă. Fiecare jeton al semaforului poate lua starea "Ocupat" sau "Liber". Un utilizator poate prelua controlul semaforului (adică poate să obțină acces la resursa comună), dacă există cel puțin un jeton liber în cadrul semaforului. Concomitent cu preluarea controlului, un jeton devine ocupat. Dacă controlul este predat, un jeton devine liber.

S-a încercat implementarea unei astfel de semaforizări în cadrul mini-sistemului CIM2000, fig.6.32, [10].

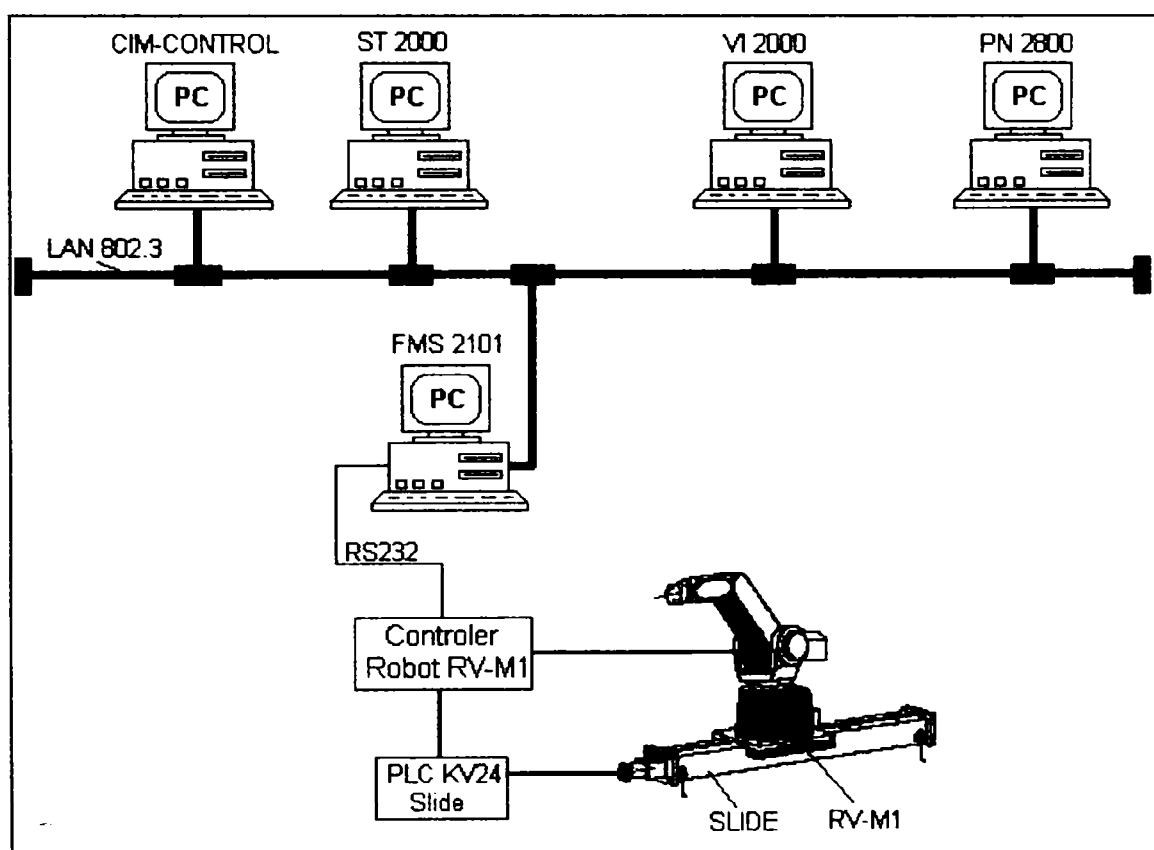


Fig.6.32. Accesul simultan de la mai multe calculatoare la controlul robotului RV-M1

S-a considerat că *resursa comună este robotul industrial RV-M1* și la un anumit moment dat mai mulți utilizatori (de exemplu studenți), de la mai multe calculatoare PC din cadrul mini-sistemului, doresc simultan să emite comenzi în vederea punerii în mișcare a robotului. Pentru a preîntâmpina apariția unor mișcări necontrolate la robot, mergând chiar până la blocarea sistemului, s-a introdus o semaforizare, cu numai un singur jeton.

Astfel, dacă un utilizator a intrat în posesia jetonului respectiv, adică *a preluat controlul semaforului*, practic a obținut accesul la resursa comună, deci putea să emite comenzi pentru realizarea diferitelor mișcări dorite. În caz contrar accesul a fost refuzat în mod automat.

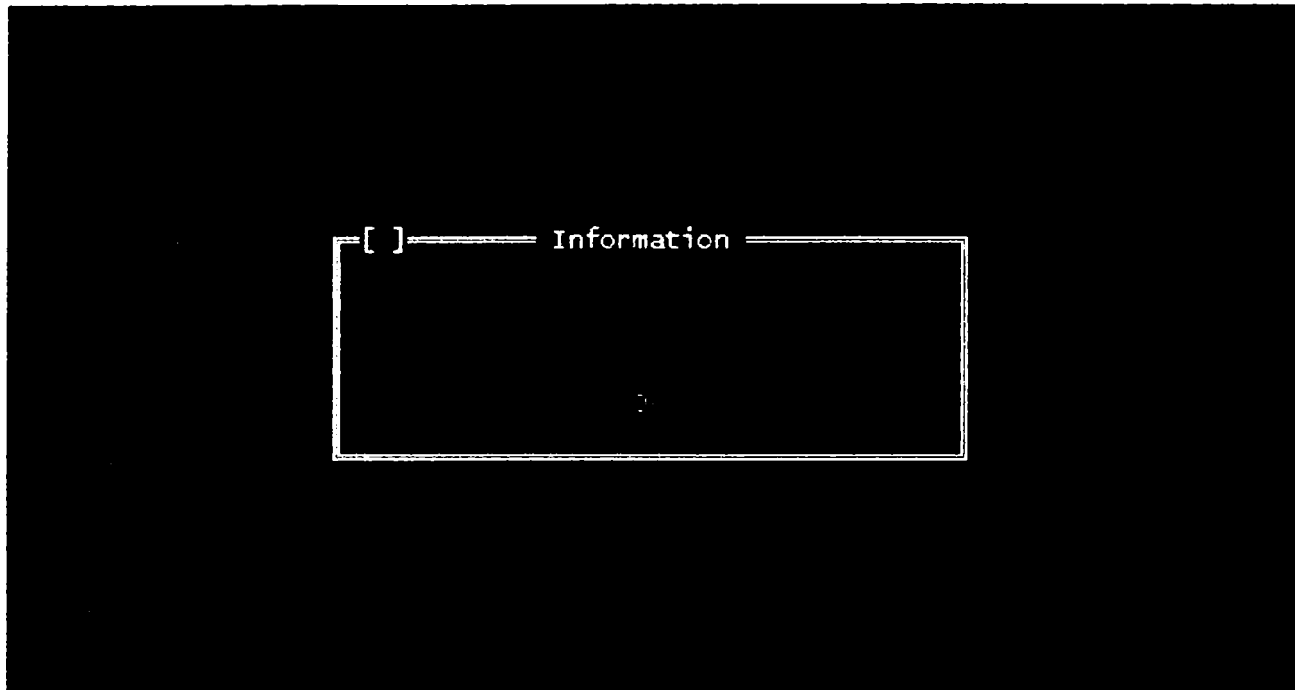


Fig.6.33. Refuzarea accesului la controlul robotului RV-M1, [10]

Jetonul a fost alocat pe rând la cele 4 calculatoare (CIM-CONTROL, ST2000, VI2000 și PN2800) disponibile din mini-sistemul CIM2000. Dacă un utilizator, care nu era în posesia jetonului, încerca totuși emiterea unor comenzi pentru comanda robotului, programul Server-Robot (de pe calculatorul FMS2101) a refuzat acest lucru, iar pe monitorul utilizatorului respectiv a apărut un mesaj care indica faptul că robotul este "ocupat" (busy), fig.6.33.

Trebuie arătat faptul că acest mod de semaforizare se poate extinde și la celelalte stații ale mini-sistemului CIM2000, cu condiția implementării și în cadrul acestora, a unor structuri deschise de comandă.

6.6. Concluzii

În acest capitol s-au prezentat principalele structuri deschise de comandă elaborate pentru mini-sistemul CIM2000. În prima fază s-a elaborat un proiect de dezvoltare a mini-sistemului CIM2000, prin care s-a prevăzut introducerea în mini-sistem a unor structuri deschise de comandă. Ca rezultat al implementării proiectului, s-a prezentat noua structură a sistemului de comandă aferentă celulei de fabricație CF2101.

În continuare s-a concentrat asupra implementării în mini-sistemul CIM2000, a modelului de accesare de la distanță a resurselor fizice și logice ale unui robot, care a fost elaborat și prezentat în capitolul 5. La implementarea acestui model trebuia ținut cont de arhitectura închisă a controlerului robotului RV-M1. S-a subliniat că această arhitectură nu permite modificarea/adaptarea structurii de comandă a controlerului, în vederea obținerii unei configurații apropiate modelului elaborat. În consecință, modelul amintit a fost implementat parțial pe calculatorul de interfață a robotului. Acest calculator pe de o parte a fost conectat la controlerul robotului RV-M1 printr-o legătură serială RS232c, iar pe de altă parte printr-o rețea LAN de tip Ethernet la un alt calculator PC, aflat pe rol de controler de celulă.

Astfel s-a obținut o structură de comandă cu arhitectură client-server, în care Client-ul era controlerul de celulă iar Server-ul era sistemul robot format din calculatorul de interfață, controler și robotul propriu-zis. Caracterul deschis al structurii de comandă era asigurat pe de o parte de platformele hardware deschise ale calculatoarelor PC, iar pe de altă parte de metoda deschisă utilizată pentru transferul mesajelor prin rețeaua LAN.

Cu ajutorul structurilor elaborate se poate accesa de la distanță unele resurse fizice și logice ale robotului RV-M1, utilizând în acest scop un set de cereri de servicii. Ele au fost concepute astfel încât să ofere un mod standard de comandă de la distanță, a oricărui robot industrial, indiferent de producător. Astfel ele stau la baza realizării unor structuri deschise de comandă în Hipersisteme CIM.

Unele structuri elaborate sunt destinate întregului mini-sistem CIM2000. Astfel de exemplu a fost elaborată o structură de semaforizare a accesului la resurse comune. Această structură a fost utilizată pentru semaforizarea accesului la comanda singurului robot industrial din mini-sistem, considerat a fi o resursă comună. Însă ea poate fi extinsă și la celelalte stații, cu condiția implementării și în cadrul acestora, a unor structuri deschise de comandă.

CAPITOLUL 7.

EXPERIMENTAREA UNOR STRUCTURI DESCHISE DE COMANDĂ PRIN CONSTRUIREA UNEI CELULE FLEXIBILE DE FABRICAȚIE INTEGRAT CU CALCULATORUL

7.1. Introducere

În capitolul 6. s-a prezentat un proiect de dezvoltare a mini-sistemului CIM2000 care prevedea introducerea în acesta, a unor de structuri "deschise" de comandă. Trebuie arătat că acest proiect s-a putut implementa doar parțial, numai pentru o parte a mini-sistemului CIM2000. Motivul este că interfețele utilizator ale echipamentelor de comandă din acest mini-sistem au caracter închis iar specificațiile aferente nu sunt disponibile. În consecință ele nu pot fi modificate sau adaptate astfel încât, să fie compatibile cu structurile de comandă deja elaborate.

Singura posibilitate a rămas crearea unor interfețe (utilizator) noi, separat pentru fiecare echipament respectiv stație. Aceasta însă, este îngreunată de lipsa documentațiilor aferente respectiv de faptul că unele echipamente de comandă din mini-sistem sunt dotate cu protocoale brevetate de comunicație, cu specificații inaccesibile. În aceste condiții, experimentarea la nivelul întregului mini-sistem CIM2000, a unor structuri deschise de comandă, a devenit practic imposibilă.

În consecință s-a luat decizia construirii unei structuri de fabricație (integrat cu calculatorul), care să fie prevăzută cu un sistem deschis de comandă. Această structură este de fapt o *celulă flexibilă de fabricație* care s-a obținut prin integrarea fizică și informațională a unor module de fabricație adecvate, respectiv existente în cadrul Universității din Oradea, fig.7.1.



Fig.7.1. Celula flexibilă de fabricație realizată

7.2. Prezentarea componentelor celulei flexibile de fabricație

Celula flexibilă de fabricație a fost construită în laboratorul AICM2 ("Automatizări în Industria Constructoare de Mașini), a Facultății de electrotehnică și în informatică, unde exista deja două mașini-unelte cu comandă numerică: un centru de prelucrare de frezare și găurire CP20UO și o mașină de frezat și găurit GPR45NC.

Cele două "*mijloace de producție*" automatizate au stat la baza construirii celulei de fabricație, fig.7.2.

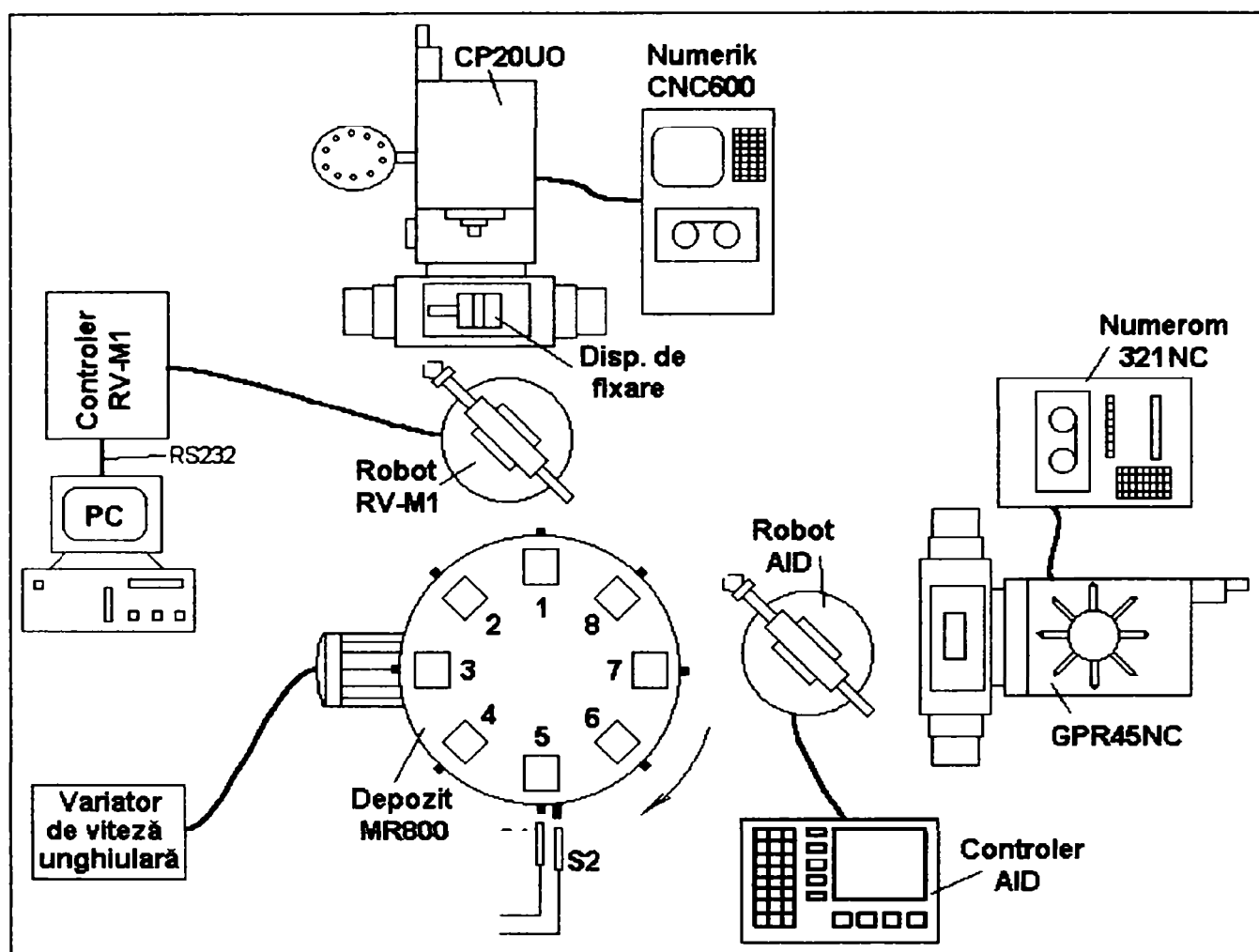


Fig.7.2. Componentele inițiale ale celulei de fabricație

- **Centrul de prelucrare** este o dezvoltare proprie a unui grup de cercetători din Universitatea Oradea. Autorul tezei a contribuit la această dezvoltare prin elaborarea proiectului de modificare/adaptare a părții mecano-hidraulice. Practic s-a plecat de la o mașină de găurit și frezat de tip GP45 NC, fabricat la fosta întreprindere "ÎNFRĂȚIREA" Oradea (actualmente STIMIN), care avea în dotare un echipament de comandă numerică de tip "NUMEROM 321", de fabricație autohtonă. Trebuie menționat faptul că mașina era prevăzută cu un sistem de comandă secvențială, a celor 4 axe de prelucrare (X, Y, Z, W), pentru acționarea cărora era utilizat numai un singur motor de curent continuu.

Între anii 1992-1994, s-a realizat transformarea acestei mașini într-un centru de prelucrare, prin atașarea la aceasta a unui magazin de scule de tip disc cu o ca-

pacitate de 20 de scule. La fel s-a modificat și partea de acționare la nivelul axelor, astfel încât prin intermediul unor motoare suplimentare de curent continuu a devenit posibilă comanda simultană a 3 axe din cele 4 (Z și W fiind axe paralele s-a menținut varianta de acționare cu un singur motor). De asemenea a fost înlocuit și echipamentul de comandă numerică, cu unul mai performant, de tip "NUMERIK CNC600", de fabricație RDG, care deja putea controla numeric 5 axe respectiv era dotat cu o interfață de comunicație de tip DNC. Aceasta din urmă a oferit posibilitatea conectării în viitor, la un calculator extern, de exemplu PC.

Tot în această perioadă s-a dotat mașina cu o masă rotativă cu comandă numerică (MR360NC), care de asemenea este o realizare practică a Universității Oradea (având la bază un proiect al întreprinderii Înfrățirea). Această masă rotativă a fost prevăzută inițial cu un dulap independent de comandă respectiv cu afișaj propriu (NUMEROM306). Dorind însă obținerea unui centru de prelucrare cu 5 axe, afișajul a fost decuplat, iar comanda motorului de acționare respectiv sistemul de măsură, a fost conectat la echipamentul CNC600. Astfel masa rotativă a devenit axa B a centrului de prelucrare.

- **Mașina-unealtă GPR45 NC** a fost realizată de întreprinderea Înfrățirea Oradea, și din construcție a fost prevăzută cu un cap revolver, cu o capacitate de 6 scule. Aceasta mașină "încă" este dotată cu un echipament de comandă numerică de tip NUMEROM 321.

Pentru **realizarea funcției de depozitare** a semifabricatelor s-a introdus un depozit tampon MR800 cu 8 posturi de lucru, fig.7.3. Acest depozit practic are la bază masa rotativă MR360NC, realizată în 1992, respectiv care era în dotarea centrului de prelucrare CP20UO. În dorința construirii unei celule de fabricație, în care să fie rezolvată și problema depozitării (a semifabricatelor respectiv a pieselor finite), masa rotativă a fost transformată într-un depozit tampon. În consecință a fost preluată de pe masa centrului de prelucrare și a fost așezată între cele două mașini.



Fig.7.3. Depozitul MR800

Cu scopul obținerii unei celule de fabricație, cu flexibilitate cât mai ridicată, s-a prevăzut ca acest depozit să utilizeze un sistem independent de comandă, adică să nu fie sub comanda echipamentului CNC600. Acest sistem de comandă, bineînțeles trebuie să aibă posibilitatea conectării informaționale, la calculatorul care va fi pe post de controler de celulă. Având în vedere că arhitectura echipamentului de comandă NUMEROM306 (cu care a fost echipată inițial masa rotativă MR360NC) nu permite această conectare, a apărut necesitatea realizării unui alt sistem de comandă.

Noul sistem de comandă trebuia să asigure un control adecvat al motorului de acționare (motor de curent continuu), via unui variator electronic de viteză unghiulară, pentru punerea în mișcare a discului depozit respectiv pentru oprirea programabilă a acestuia (cu o precizie suficientă), în cele 8 poziții aferente posturilor de lucru. Pentru facilitarea obținerii pentru acest depozit, a unui sistem de comandă cât mai simplu, s-a renunțat la sistemul de măsură cu traductor inductosin circular și s-a dotat depozitul cu o serie de came respectiv cu 2 senzori de proximitate (notați cu S1 și S2 în fig.7.2).

Pentru **realizarea funcției de transfer** semifabricate și piese finite, între depozit și mijloacele de producție, s-au introdus 2 roboți industriali. Unul a fost preluat din laboratorul CIM (robotul industrial RV-M1) împreună cu calculatorul PC de interfață. Celălalt robot (AID) a fost împrumutat din laboratorul de Roboți industriali al Facultății de tehnologie și inginerie managerială. Acesta din urmă este un robot industrial cu 6 grade de mobilitate cu acționare electrică. Este utilizat pentru servirea (cu semifabricate) a mașinii GPR45NC.



Fig.7.4. Robotul RV-M1 în așteptarea terminării operației de prelucrare pe CP20UO

Robotul industrial RV-M1 a fost utilizat pentru servirea (cu semifabricate) a centrului de prelucrare CP20UO (figurile 7.4. și 7.5), iar robotul AID pentru servirea mașinii GPR45NC (fig.7.6.).

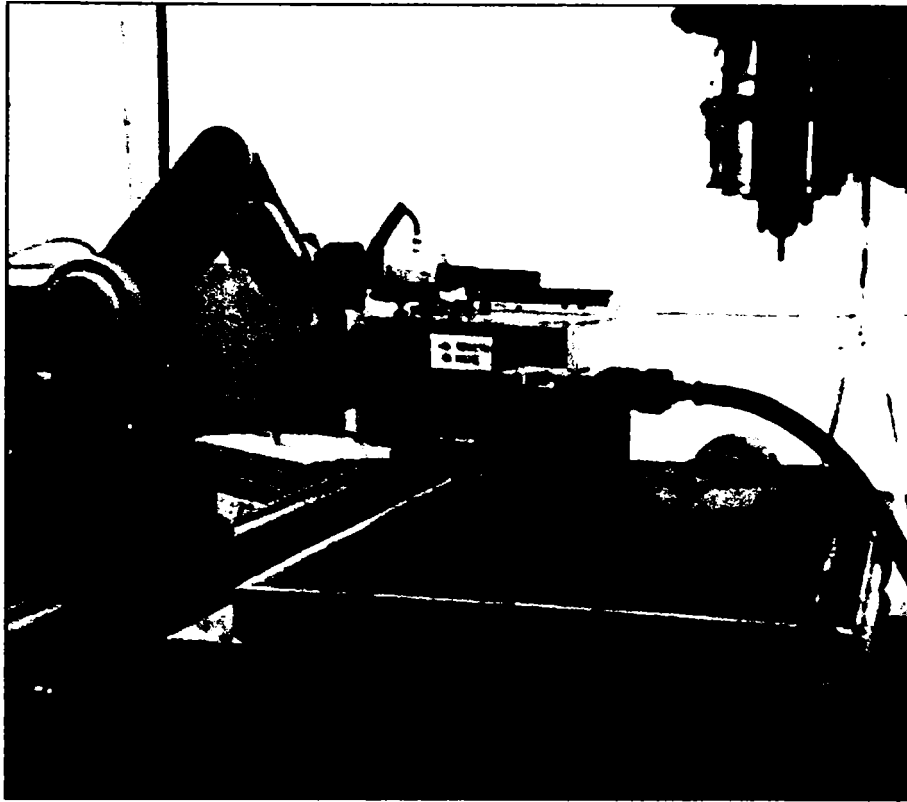


Fig.7.5. Robotul RV-M1 realizând preluarea piesei din dispozitivul de fixare

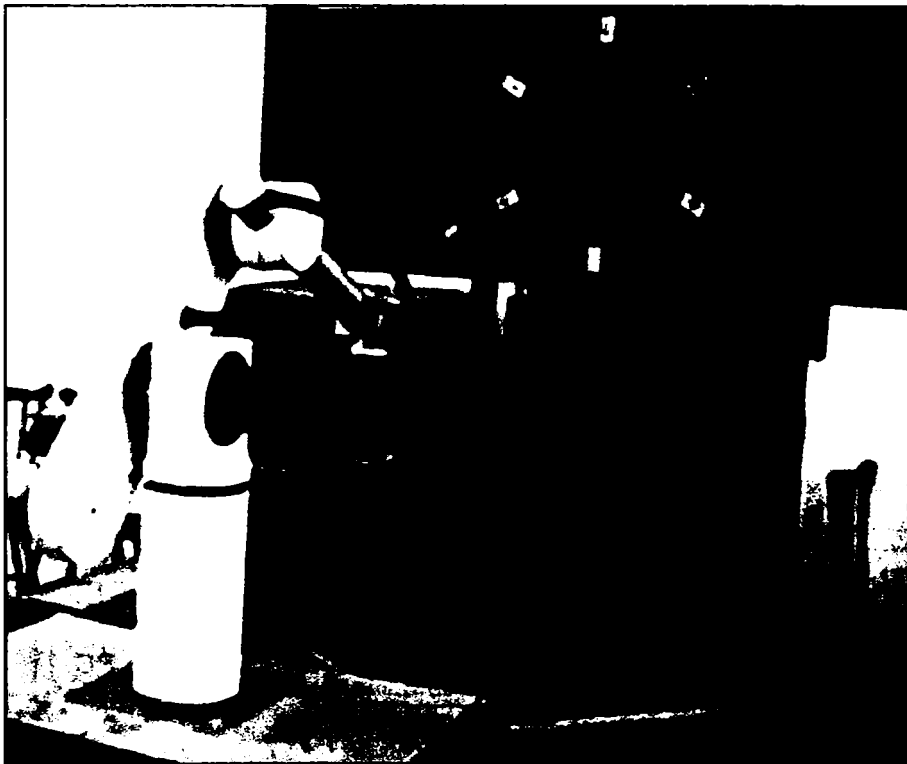


Fig.7.6. Robotul AID în servirea mașinii GPR45NC

Celula de fabricație, cu componentele prezentate a fost destinată prelucrării prin așchiere a semifabricatelor prismatice care inițial sunt așezate manual, de către un operator uman, în cele 8 posturi ale depozitului rotativ. Pentru ca această celula de fabricație să funcționeze automat, fără intervenția operatorului uman (cel puțin pentru 8 piese), trebuia realizată integrarea informațională a componentelor prezentate într-un sistem unitar, cu ajutorul calculatorului. În acest scop trebuia concepută structura sistemului de comandă.

7.3. Conceperea și realizarea sistemului de comandă

Arhitectura sistemului de comandă a celulei de fabricație a fost concepută astfel încât să permită implementarea unor structuri deschise (de comandă). Astfel s-a prevăzut conectarea "serială" a fiecărui echipament de comandă (NC, CNC, ROC) cu câte un calculator PC (fig.7.7.), apoi interconectarea acestor calculatoare într-o rețea LAN de tip Ethernet. Pentru comanda/coordonarea activităților de fabricație la nivelul celulei s-a prevăzut introducerea unui calculator PC pe rol de controler de celulă. Acest calculator a fost de asemenea conectat la rețeaua LAN.

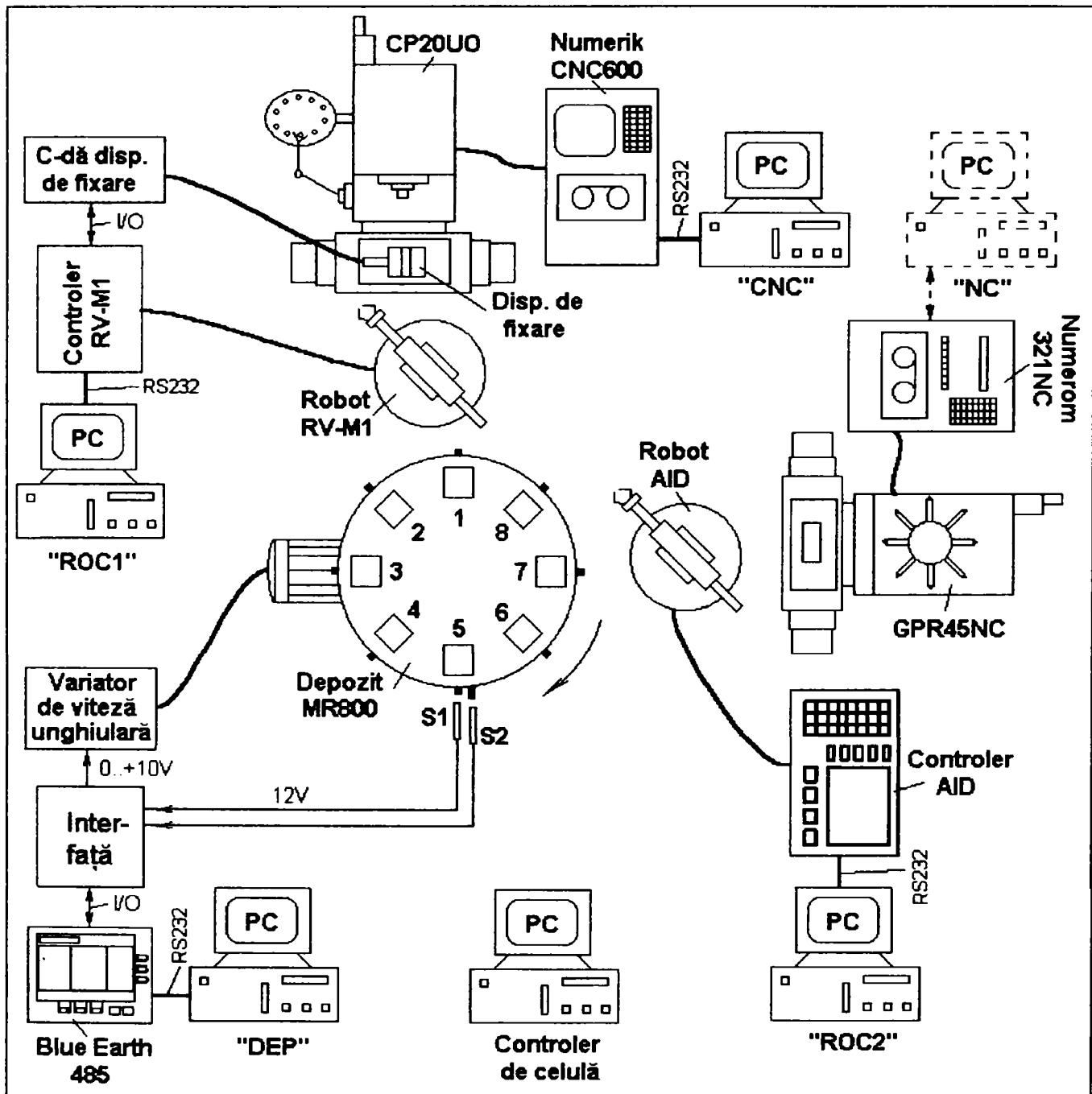


Fig.7.7. Interfațarea cu calculatoare PC a echipamentelor de comandă

În cazul depozitului tampon MR800 mai trebuia concepută și realizată în plus un sistem de comandă locală. Pentru acesta s-a utilizat un microcalculator Blue Earth 485 respectiv s-a realizat o interfață hardware. Interfața realiza conectarea la microcalculator a variatorului de viteză unghiulară și a senzorilor de proximitate.

Trebuie menționat că interfața I/O a microcalculatorului Blue Earth 485 nu este dotată cu ieșiri analogice (doar cu intrări analogice respectiv cu intrări și ieșiri numerice), [16]. Ca urmare pentru comanda variatorului, care necesita un semnal analogic în domeniul 0...+10V, trebuia realizată și o interfață hardware, care avea rolul unui convertor numeric-analog (DAC – Digital Analog Converter). La valoarea de 0V a semnalului, motorul este oprit, iar la valoarea de +10V se obține turația maximă într-un sens. Modificând nivelul semnalului în domeniul respectiv, se obține o turație mai mică sau mai mare la motor, și în același timp la discul depozit.

S-a considerat că e suficientă realizarea **rotirii discului depozit numai într-un singur sens**. Ca atare nu era necesară utilizarea unor semnale de comandă în domeniul negativ: -10V...0V aferente sensului celălalt de rotire. Adoptarea acestei variante de comandă cu rotire numai într-un singur sens, avea avantajul obținerii unei interfețe mai simple.

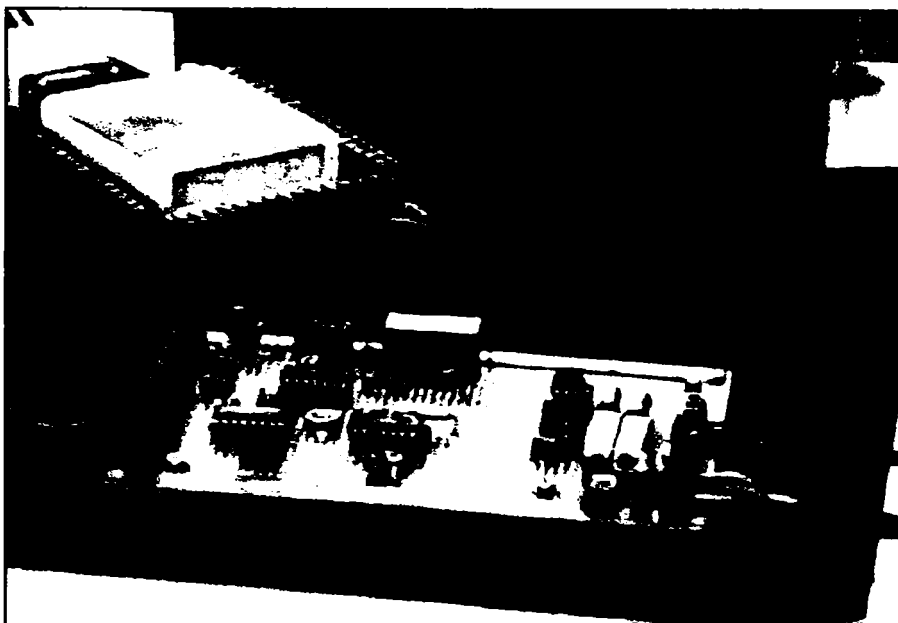


Fig.7.7. Microcalculatorul Blue Earth 485 și interfața cu variatorul

Interfața realizată mai avea și rolul alimentării (cu o tensiune de 12Vcc), a senzorilor de proximitate, respectiv preluarea și adaptarea semnalelor furnizate de aceștia astfel încât nivelul lor să fie compatibile cu cele suportate de porturile de intrare ale microcalculatorului (5V TTL).

Microcalculatorul Blue Earth 485 fiind dotat din fabricație cu interfață serială RS232c, s-a putut conecta la un calculator PC, [16]. Astfel s-a putut rezolva atât problema comenzii locale cât și conectarea informațională, printr-un calculator PC de interfață, la controlerul de celulă.

Singura problemă care nu s-a putut rezolva până în prezent este problema conectării echipamentului de comandă numerică NUMEROM321 la un calculator PC. Acest echipament NC, fiind de tip mai vechi, nu este dotat cu o interfață de comunicație (de exemplu serială) care să permită conectarea acestuia la un alt echipament sau calculator. În aceste condiții, în cadrul experimentărilor efectuate, nu s-a putut emite comenzi de la distanță (adică de la controlerul de celulă), pentru pomirea operației de prelucrare pe mașina GPR45NC. Însă robotul AID realiza totuși încărcarea și descărcarea mașinii cu semifabricate.

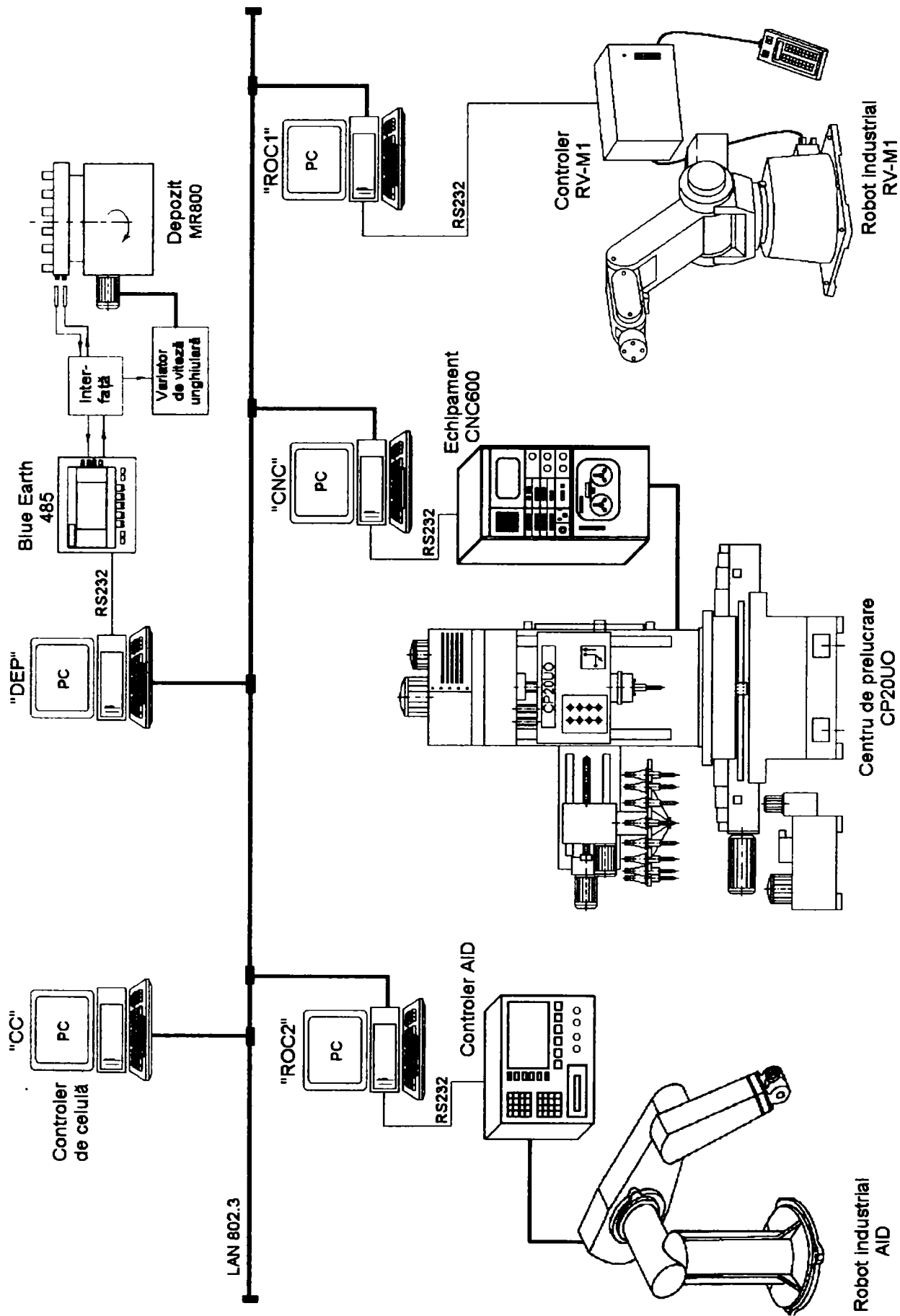


Fig. 7.8. Interconectarea calculatoarelor PC de interfață și a controlerului de celulă

Celelalte echipamente de comandă (controlerul RV-M1, controlerul AID, microcalculatorul Blue Earth 485 respectiv echipamentul CNC600), fiind dotate cu interfețe seriale RS232c, nu prezentau astfel de probleme hardware de conectare la calculatoare PC. În aceste condiții, după interconectarea calculatoarelor PC într-o rețea locală LAN, a rezultat pentru sistemul de comandă o arhitectură prezentată în fig.7.8.

După realizarea interconectărilor fizice din fig.7.8., s-a trecut la **implementarea structurilor software de comandă**. Aceasta s-a efectuat în două etape:

1. În prima etapă s-a realizat, separat pentru fiecare echipament de comandă, **conectarea informațională la calculatorul PC** de interfață,
2. iar în a doua etapă s-a realizat **conectarea informațională la controlerul de celulă** a calculatoarelor PC de interfață.

Pentru conectarea informațională la calculatoare PC, a echipamentelor de comandă din cadrul celei de fabricație, era necesară realizarea unor programe de comunicație serială. Aceste programe (rulate pe calculatoarele PC), aveau implementate:

- **protocoalele specifice (brevetate)** ale fiecărui echipament în parte, aferente realizării *schimburilor informaționale* prin legătură serială (transmiterea comenzilor și a programelor de aplicație);
- **interfețe utilizator** pentru comandă locală (necesară pentru inițializare respectiv pentru efectuarea unor eventuale intervenții manuale, de către un operator).

În a doua etapă, aceste programe de comunicație serială au fost completate cu **protocoale aferente transmiterii comenzilor prin rețea LAN** respectiv cu **funcțiile server** care asigurau conectarea informațională la controlerul de celulă

În cazul controlerului robotului RV-M1, un astfel de program de conectare la un calculator PC respectiv la controlerul de celulă, exista deja de la implementările efectuate în cadrul mini-sistemului CIM2000 (prezentate în capitolul 6.). La celelalte echipamente însă, programele respective trebuiau să fie concepute și realizate. Pentru reducerea volumului de muncă necesar realizării acestor programe, s-a plecat de la interfața utilizator dezvoltată anterior pentru robotul RV-M1 și s-a încercat adaptarea acesteia, cu mici modificări, și pentru celelalte echipamente. Bineînțeles partea de comunicație serială, care necesita implementarea unor protocoale specifice (brevetate), era diferită în cazul fiecărui echipament în parte.

7.3.1. Conectarea informațională a echipamentului CNC al centrului de prelucrare CP20UO la un calculator PC

Cea mai complicată a fost conectarea informațională la un calculator PC a echipamentului CNC600, care are implementat un protocol de transmitere date de tip master-slave, [112]. Pe baza acestuia, întâi atât calculatorul PC cât și CNC-ul sunt considerate două stații "echivalente" d.p.d.v. al comunicațiilor, astfel încât oricare din ele pot iniția realizarea unor schimburi informaționale.

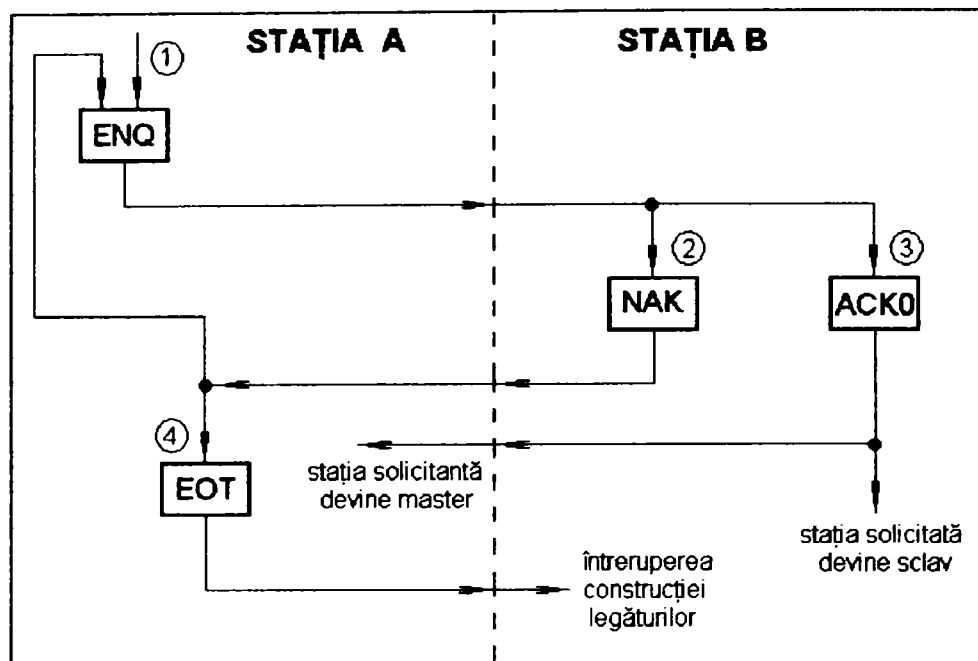


Fig.7.9. Protocolul simplificat al inițierii comunicației, [112]

Conform fig.7.9., dacă una dintre stații (de exemplu A) dorește realizarea unei transmisi de date, atunci ea trebuie să emite mai întâi către cealaltă stație, codul corespunzător unei cereri de apel notat cu ENQ (Enquiry). Dacă stația solicitată nu este gata pentru recepție (2), atunci emite înapoi un răspuns negativ NAK (Negativ ACK). În acest caz stația inițitoare repetă solicitarea de 3 ori, după care întrerupe legătura (4), prin EOT (End of Transmission – Sfârșit transmisie). Dacă stația solicitată este gata pentru recepție (3), atunci emite un răspuns pozitiv la solicitare (ACK – Acknowledge). Ca urmare stația solicitată preia **starea de recepție (SLAVE)**, iar stația solicitantă preia **starea de emisie (MASTER)**.

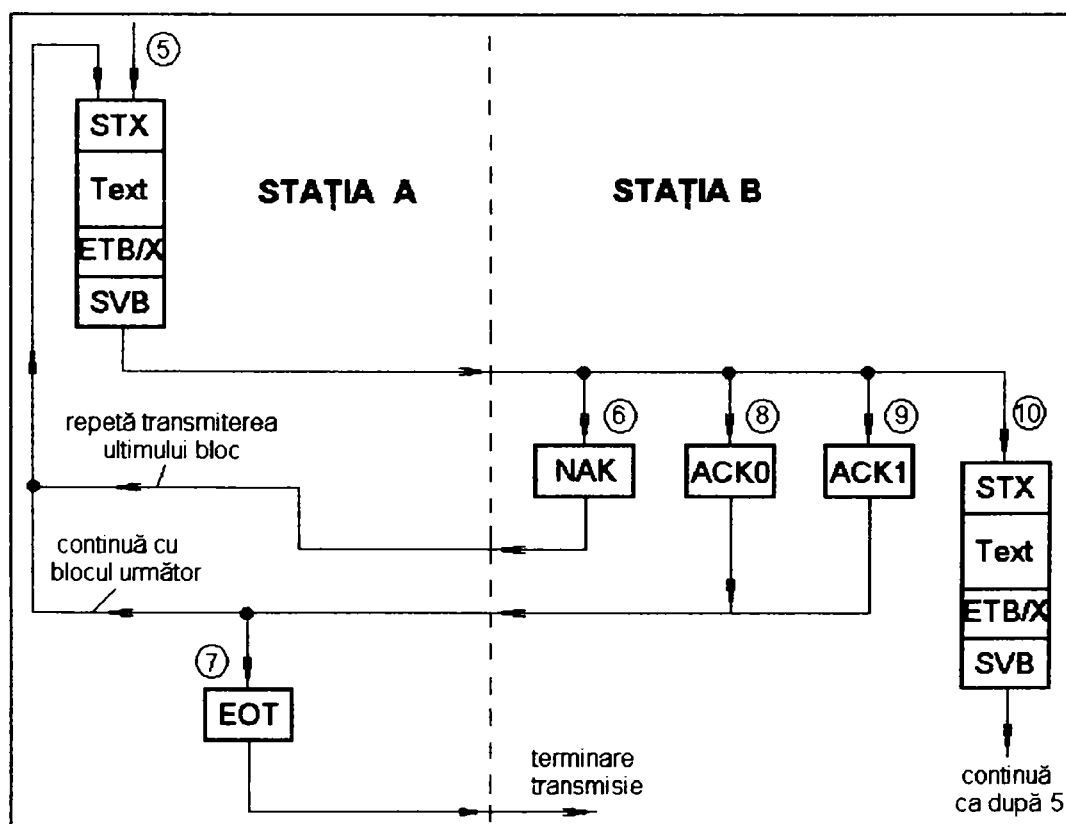


Fig.7.10. Protocolul simplificat al transmisiiei datelor, [112]

După aceasta stația Master (stația A), începe **transmiterea datelor sub forma unor blocuri** de 64 de caractere, fig.7.10. Fiecare bloc (5), este început cu un simbol STX (Start of Text), după care urmează datele propriu-zise sub forma unui șir de caractere ("Text"). Blocul este terminat cu un simbol ETB (End of Block) sau cu ETX (End of Text) în cazul ultimului bloc, urmat de o secvență de verificare bloc (SVB) numit și "sumă de control".

Suma de control este verificată de stația Slave, care în caz de eroare răspunde negativ cu NAK (Negativ ACK), iar stația de emisie repetă transmiterea ultimului bloc (6). În cazul unor sume de control corecte stația receptoare răspunde pozitiv la primul și la următoarele blocuri impare cu ACK0. La blocurile pare răspunde cu ACK1. După ultimul bloc transmis stația emițătoare întrerupe legătura cu un simbol EOT (End of Transmission).

Dacă blocul care s-a transmis (încheiat cu ETX) conținea o solicitare pentru transmiterea unei alte informații (de ex. a unui program), atunci conform unui dialog prescurtat, stația receptoare răspunde direct cu datele solicitate (tot sub forma unor blocuri), (10). În acest caz stațiile își schimbă de fiecare dată starea de emisie-recepție.

Caractere de control transmisie	Codul în zecimal	Semnificație
ACK0	#16 + #48	Răspuns pozitiv 0
ACK1	#16 + #49	Răspuns pozitiv 1
ENQ	#5	Enquiry (Apel)
STX	#2	Start of Text (Start text)
ETX	#3	End of Text (Sfârșit text)
ETB	#23	End of Block (Sfârșit bloc)
EOT	#4	End of Transmission (Sfârșit transmisie)
NAK	#15	Răspuns negativ

Tab.7.1. Lista principalelor caractere de control al transmisiei de date, [112]

În tab.7.1 s-au prezentat codurile principalelor **caractere de control al transmisiei** de date. Pe lângă aceste caractere mai există și o altă categorie, a **caracterelor de comandă** care sunt transmise în zona "Text" a blocului de date. Cu ajutorul acestora, stația emițătoare solicită de la stația receptoare efectuarea unei operații (de exemplu încărcarea/descărcarea unui program de aplicație, selectarea și lansare în execuție etc.).

De exemplu, în cazul încărcării unui program de aplicație de la un calculator PC, se utilizează caracterul de comandă SCI (Start Cititor) prin care este solicitată de la CNC o operație de "introducere date". Formatul blocului este următorul:

STX	%9	SCI	ETB	SVB
-----	----	-----	-----	-----

unde %9 este numele programului de aplicație. După acceptarea operației de către CNC (prin emiterea unui ACK), calculatorul începe transmiterea programului, introducând frazele acestuia în zona "Text" a blocurilor de date (fără SCI).

În tabelul 7.2. s-au prezentat codurile principalelor caractere de comandă, utilizate în cadrul protocolului de transmitere date. Trebuie menționat că unele caractere de comandă pot fi utilizate de ambii parteneri de comunicație, iar altele numai de către echipamentul CNC sau de către calculatorul PC.

Caractere de comandă	Codul în zecimal	Semnificație
SCI	#99	Start cititor (introducere date)
SPE	#101	Start perforator (preluare date)
STP	#96	Start program de prelucrare
STO	#97	Stop program de prelucrare
SEP	#105	Selectare program de prelucrare
CHN	#106	Căutare frază
AFS	#108	Afișare
COR	#110	Corectare
IMM	#116	Memorare
STG	#117	Ștergere
SCR	#118	Solicitare ecran
SME	#60	Start mesaj

Tab.7.2. Lista principalelor caractere de comandă

Pe baza protocolului descris anterior, s-a realizat un program de comunicație între un calculator PC și echipamentul CNC600, cu interfața utilizator prezentată în fig.7.11.

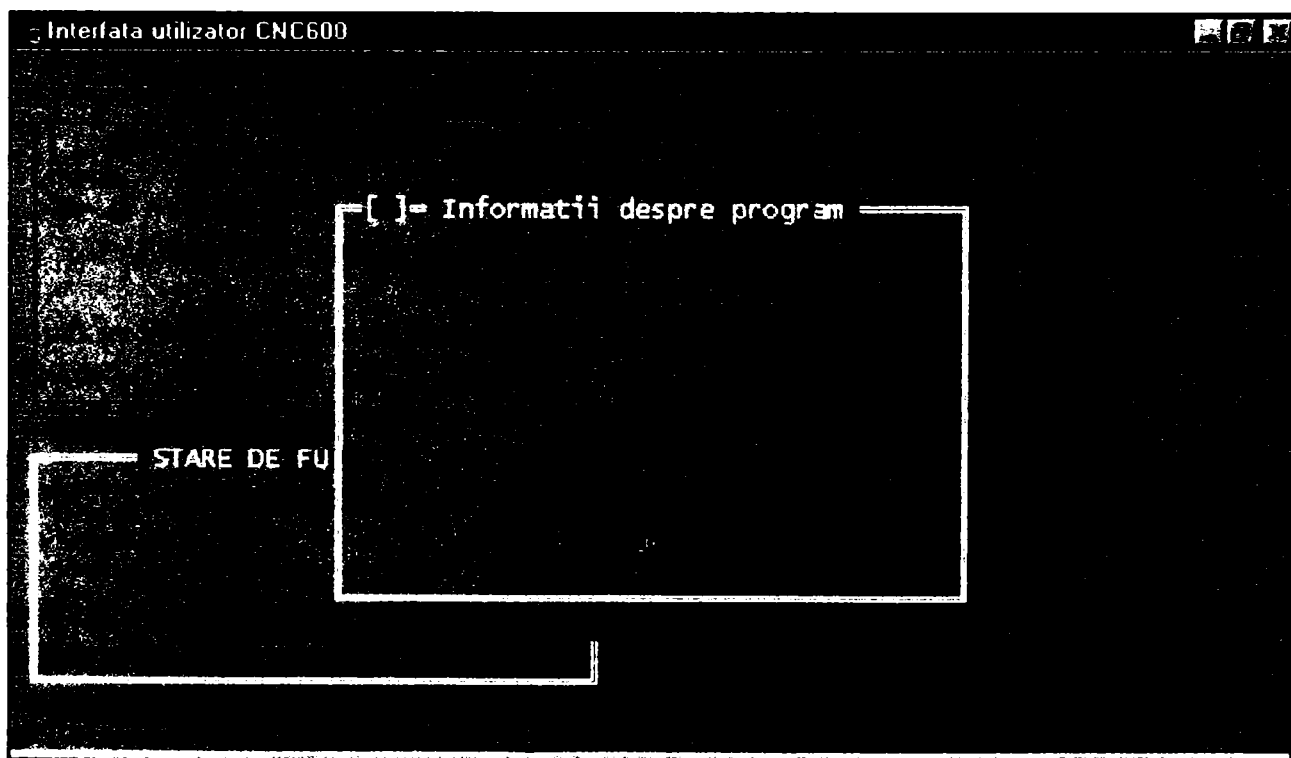


Fig.7.11. Interfața utilizator a programului de comunicație

Programul este rulat pe un calculator PC care este conectat la echipamentul CNC600 prin intermediul unui port serial RS232c. Interfața utilizator a programului realizat permite două moduri de funcționare: modul operator și modul server.

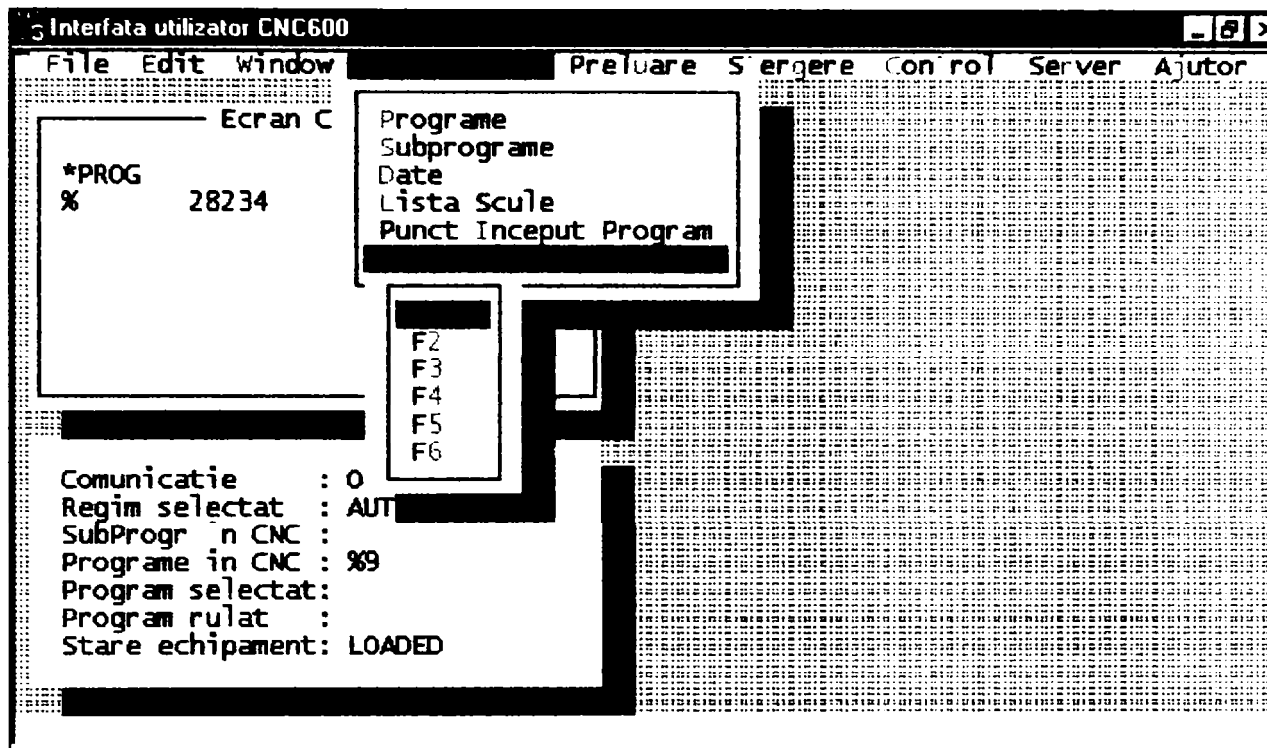


Fig.7.12. Fereastra cu submeniurile funcției de transmitere (încărcare în CNC)

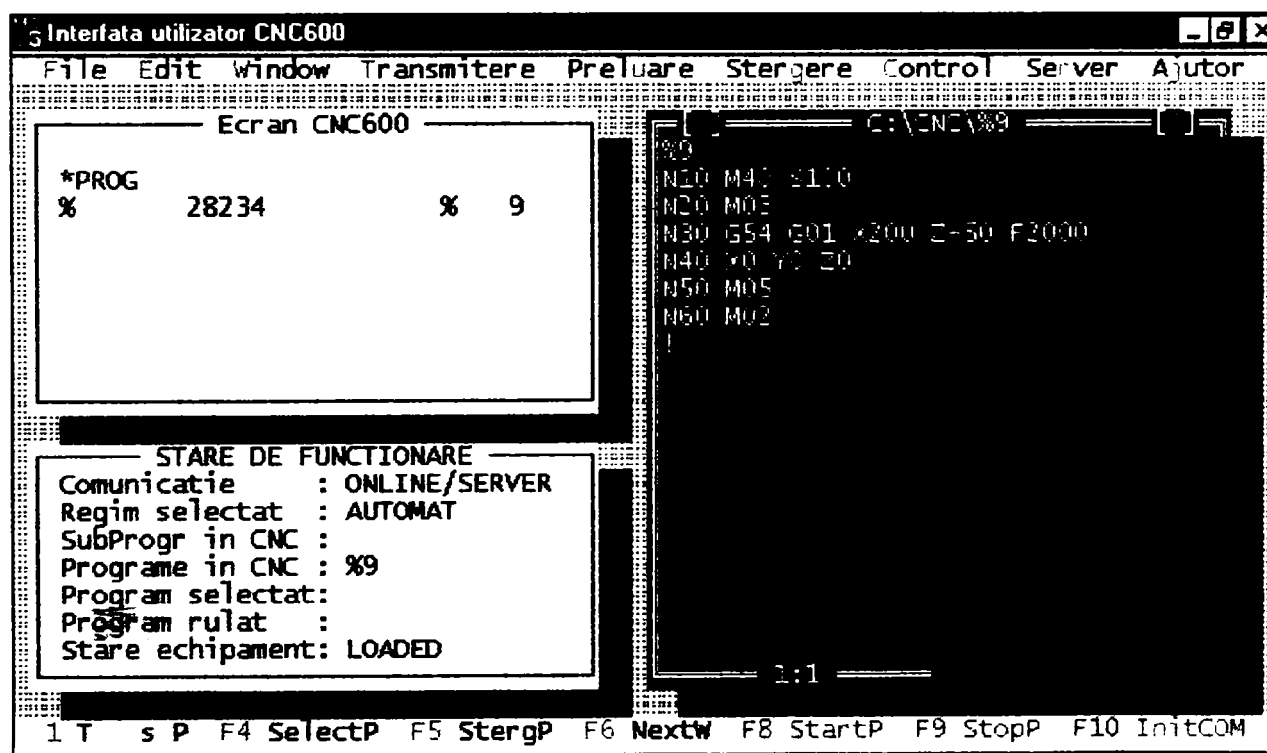


Fig.7.12. Ferestrele Ecran CNC, Stare de Funcționare, Editor de texte

În modul operator (implicit), un utilizator poate efectua următoarele operații:

- **încărcarea/descărcarea/ștergerea** în/din CNC a programelor de aplicație, a subprogramelor de prelucrare, a listelor de scule, a coordonatelor "punctelor fixe" (care indică originile sistemelor de coordonate utilizator), a coordonatelor "punctului de început program" și a parametrilor utilizator ("date"), fig.7.12,
- **selectarea, lansarea în execuție, oprirea sau continuarea execuției** unui program de aplicație existent în CNC (prin meniul "Control"),

- **preluarea și vizualizarea conținutului ecranului CNC** (este util la monitorizarea funcționării atunci când calculatorul este plasat la o distanță mai mare de la CNC), fig.7.12.,
- **urmărirea stărilor de funcționare**, legate de comunicație, de regimul de lucru selectat, de programele și subprogramele încărcate în CNC, etc., fig.7.12.
- **editarea programelor de aplicație**, cu ajutorul unui editor de texte încorporat, fig.7.12.

Modul server este o extensie a programului de comunicație serială cu CNC. Se activează prin meniul "Sever" și asigură recepția automată a cererilor de servicii emise prin rețeaua Ethernet de controlerul de celulă și execuția comenzii aferente (încărcare program, selectare, lansare etc.). De asemenea informează pe controlerul de celulă despre stările curente de funcționare, care sunt vizualizate în fereastra "STARE DE FUNCȚIONARE". În modul server nu sunt acceptate comenzi de la un operator (în afara celei de revenire din mod server în mod operator)

7.3.2. Conectarea informațională a microcalculatorului de comandă al depozitului MR800 la un calculator PC

Blue Earth 485, utilizat pentru comanda depozitului MR800, este un microcalculator de uz general, care are la bază **microcontroler-ul 80C51**, [16]. A fost achiziționat de Universitatea din Oradea în mai multe exemplare, pentru comanda diferitelor standuri didactice de acționare cu motoare electrice respectiv pneumatice, de putere mică. Între altele este dotat cu două porturi numerice (digitale), fiecare cu câte 8 linii distincte, respectiv cu un port cu 4 linii de intrare analogică.

În principiu permite încărcarea și rularea programelor utilizator scrise în limbaj de asamblare sau în **limbaj BASIC**. Pentru încărcarea programelor, de pe un calculator PC (la care este conectat printr-un **port serial RS232c**), necesită un program special de comunicație oferit separat, de firma producătoare. Acest program, datorită prețului relativ ridicat, nu a fost achiziționat de Universitatea din Oradea. Însă, conform documentației, în lipsa acestuia se poate utiliza orice alt program de tip "terminal" pentru editarea programelor în mod "linie de comandă". Un astfel de program este de exemplu "TERM95.EXE" al pachetului de programe "Norton Commander".

Utilizarea programelor de tip "terminal" însă oferă o serie de dezavantaje. De exemplu este accesibil doar **interpretorul BASIC încorporat**, în consecință programele trebuie să fie realizate în acest limbaj (BASIC), iar viteza de rulare este relativ mică (față de cele scrise în limbaj de asamblare). **Un alt dezavantaj** este faptul că programele "terminal" nu permit realizarea unui schimb automat de date, adică fără prezența operatorului uman (ele au fost concepute astfel încât să fie interactive). De exemplu un operator trebuie să introducă de la tastatură comenzile aferente încărcării frază cu frază a programului de aplicație după care lansează în execuție cu comanda "RUN". La oprire (sfârșit execuție), pentru relansarea programului este nevoie iarăși de prezența operatorului.

Având în vedere că **sistemul de comandă al depozitului** trebuie să ofere, pe lângă modul operator și modul server (caracterizat printr-un schimb automat de in-

formații și date), s-a luat decizia realizării unui program adecvat de comunicație. Protocolul care trebuia implementat era relativ simplu: calculatorul PC transmite o comandă sau o frază de program iar microcalculatorul răspunde printr-un mesaj de acceptare sau în caz de eroare de sintaxă: cu un mesaj de eroare. Astfel încărcarea unui program de aplicație se rezumă la încărcarea frază cu frază a acestuia. Pentru lansare în execuție se utilizează comanda "RUN" iar pentru ștergerea memoriei (în vederea încărcării unui alt program) comanda "NEW", [16].

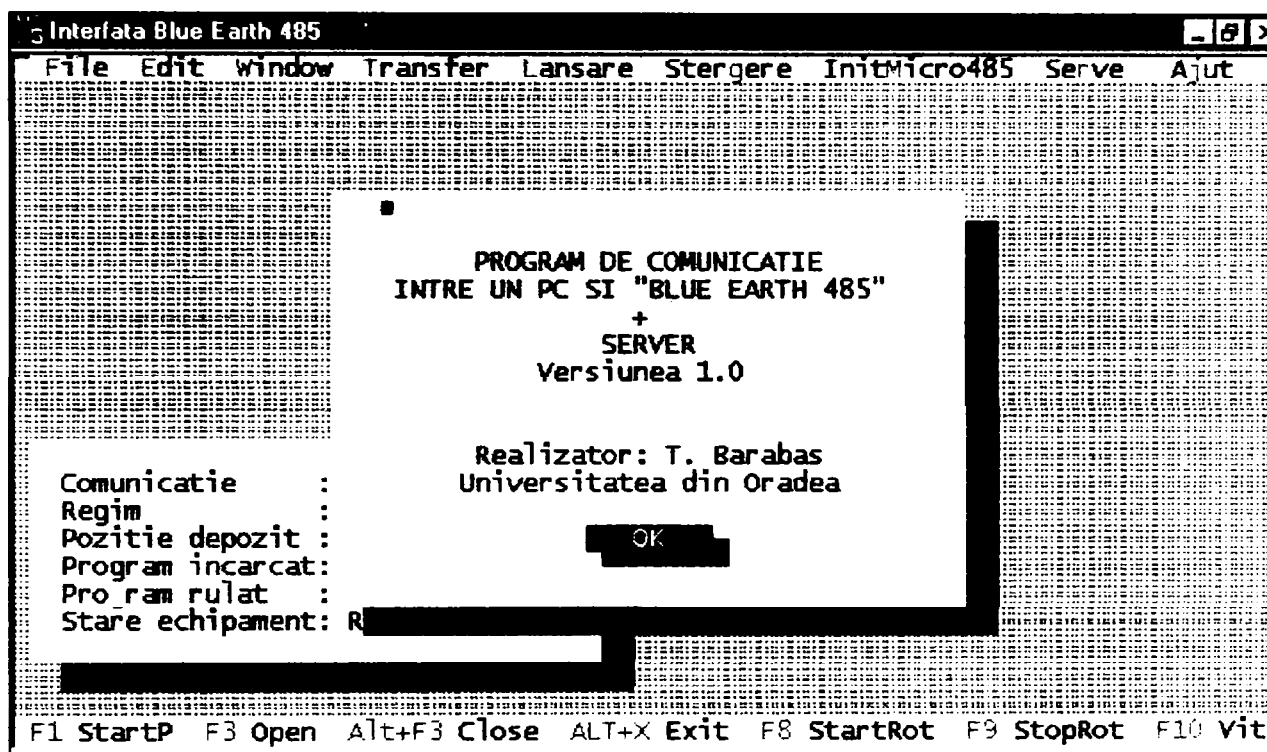


Fig.7.13. Interfața utilizator a programului de comunicație

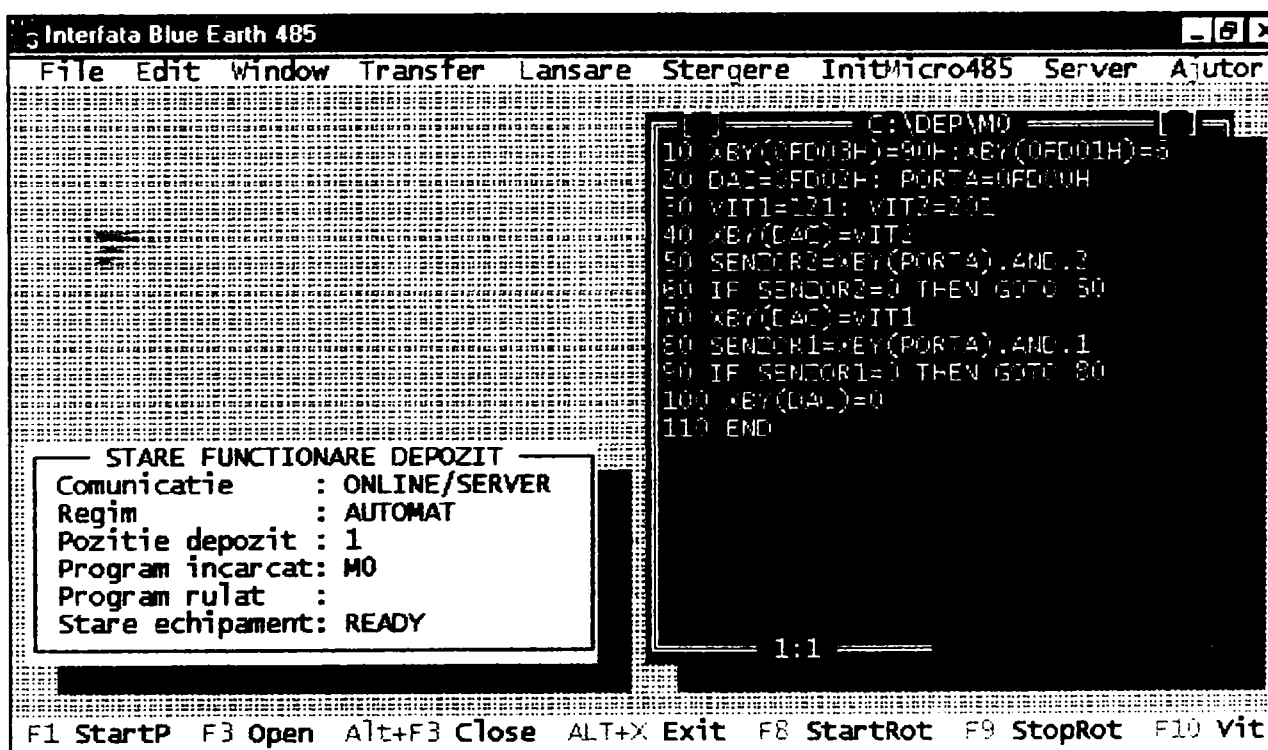


Fig.7.14. Fereastra "Stare Funcționare Depozit" și Editorul de texte

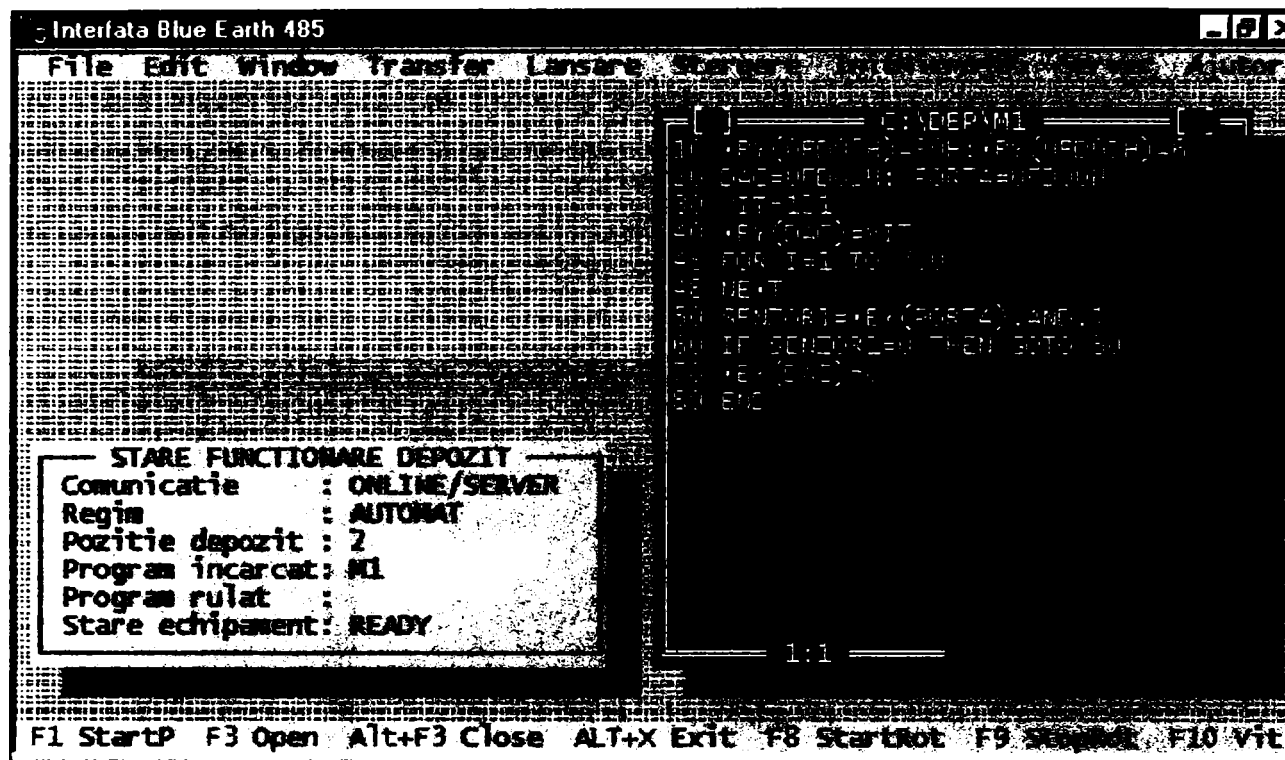


Fig.7.15. Lansarea programului M1 cauzează incrementarea poziției discului depozit

Programul realizat (fig.7.13.), permite **încărcarea/descărcarea programelor** de aplicație în/din microcalculator respectiv **lansarea în execuție** și la nevoie **ștergerea** lor. La un moment dat poate fi încărcat în memoria microcalculatorului doar un singur program, în consecință nu este necesară selectarea programului înainte de lansare în execuție. Asemănător interfețelor utilizator prezentate anterior, programul are încorporat un **editor de texte** pentru crearea și/sau modificarea programelor de aplicație, fig.7.14. De asemenea utilizează o fereastră dedicată vizualizării stărilor de funcționare. În aceasta fereastră se afișează și **poziția discului magazin** (din cele 8 posibile).

În **regim manual** rotirea discului magazin poate fi controlată cu ajutorul tastelor funcționale F8 și F9 iar cu tasta F10 se poate prescrie treapta vitezei de rotație printr-o valoare în domeniul 0 și 255 (valoarea implicită este 100). Pentru controlul rotirii discului magazin în **regim automat**, se utilizează două programe "de comandă": M0 și M1, scrise în limbajul BASIC al microcalculatorului. Programul M0 (fig.7.14), se utilizează la punerea sub tensiune, atunci când poziția discului magazin nu este cunoscută sau ori de câte ori este necesară readucerea acestuia în poziția de origine. Celălalt program, comandă o mișcare de rotire cu "un pas" față de poziția curentă. Astfel, lansând o dată programul M1 (fig.7.15.), discul magazin este rotit de exemplu din poziția 1 în poziția 2.

În cadrul celor două programe, pornirea mișcării de rotație se realizează prin programarea unei anumite tensiuni de comandă (în domeniul 0 și +10V) la intrarea variatorului, care pune în mișcare motorul de acționare, fig.7.16. Având în vedere că microcalculatorul Blue Earth 485 nu este dotat cu ieșire analogică, se utilizează portul C pentru emiterea unei valori numerice pe 8 biți (corespunzător celor 8 linii de ieșire). Această valoare numerică este aplicată convertorului numeric analog (DAC) de pe interfața realizată, care o transformă în semnal analogic. Acest semnal apoi este aplicat pe bornele variatorului.

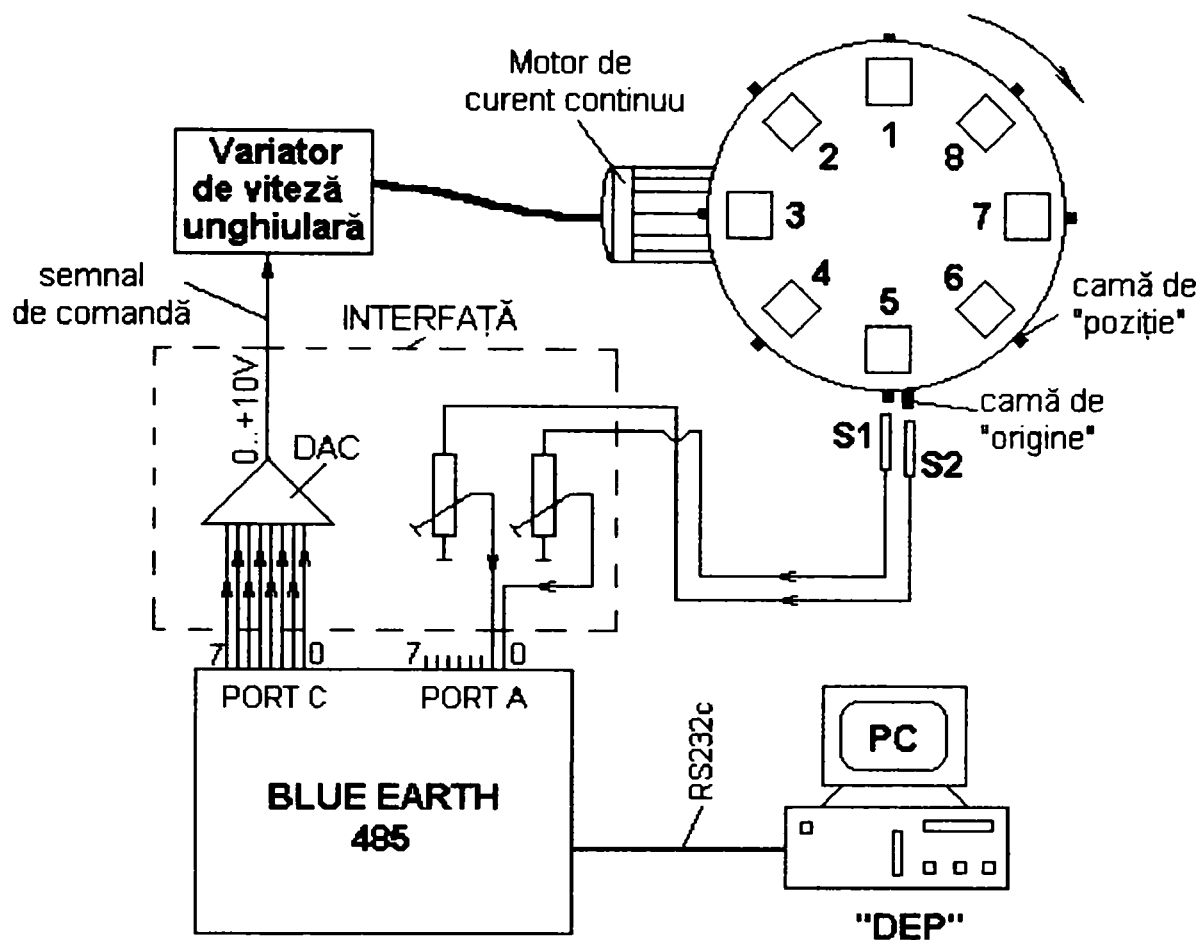


Fig.7.16. Schema de principiu a sistemului de comandă al depozitului MR800

Comanda prin care se prescrie valoarea vitezei de rotație este $XBY(DAC)=VIT$ unde variabila DAC conține adresa portului C iar VIT conține valoarea treptei de viteză de rotație. Dacă valoarea prescrisă este diferită de 0 atunci se obține o mișcare de rotație cu o viteză proporțională cu aceasta. Pentru oprirea mișcării se programează valoarea 0 la care corespunde o tensiune de comandă de 0V la variator.

În cazul **programului M1** mișcarea este oprită la primul impuls recepționat de la senzorul S1. Acest senzor (conectat la linia 0 a portului A) emite câte un impuls la fiecare din cele 8 poziții ale discului de depozit. Ca urmare dacă mișcarea are loc până la apariția primului impuls la S1 atunci se execută exact un pas (trecerea dintr-o poziție în imediat următoare). În cadrul programului se utilizează variabila SENZOR1 pentru urmărirea nivelului logic al semnalului de la linia 0 a portului A. Dacă valoarea este 1 adică a apărut impulsul la senzorul S1, atunci se comandă oprirea motorului prin $XBY(DAC)=0$.

În cazul **programului M0** mișcarea este oprită tot de impulsul dat de senzorul S1 dar numai după recepționarea de la senzorul S2 (SENZOR2) a unui "impuls de origine". În cadrul acestui program practic se utilizează două viteze: o viteză normală (VIT1) și o viteză "rapidă" (VIT2). La recepționarea impulsului de origine, viteza este redusă iar poziționarea (urmată de oprire) se realizează cu viteză normală.

Acest sistem de poziționare, în cadrul experimentărilor efectuate s-a dovedit a fi suficient de precis pentru efectuarea în condiții bune a operațiilor de transfer semifabricate.

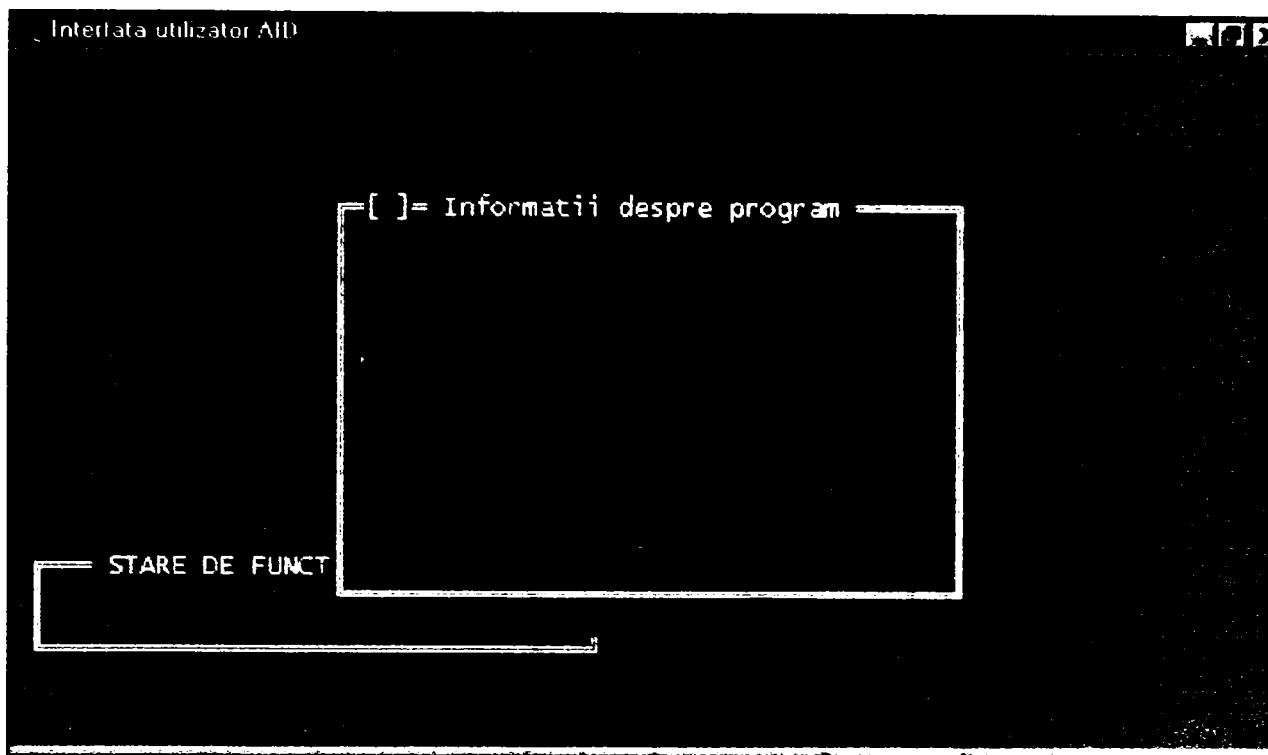


Fig.7.19. Interfața utilizator a programului de comunicație

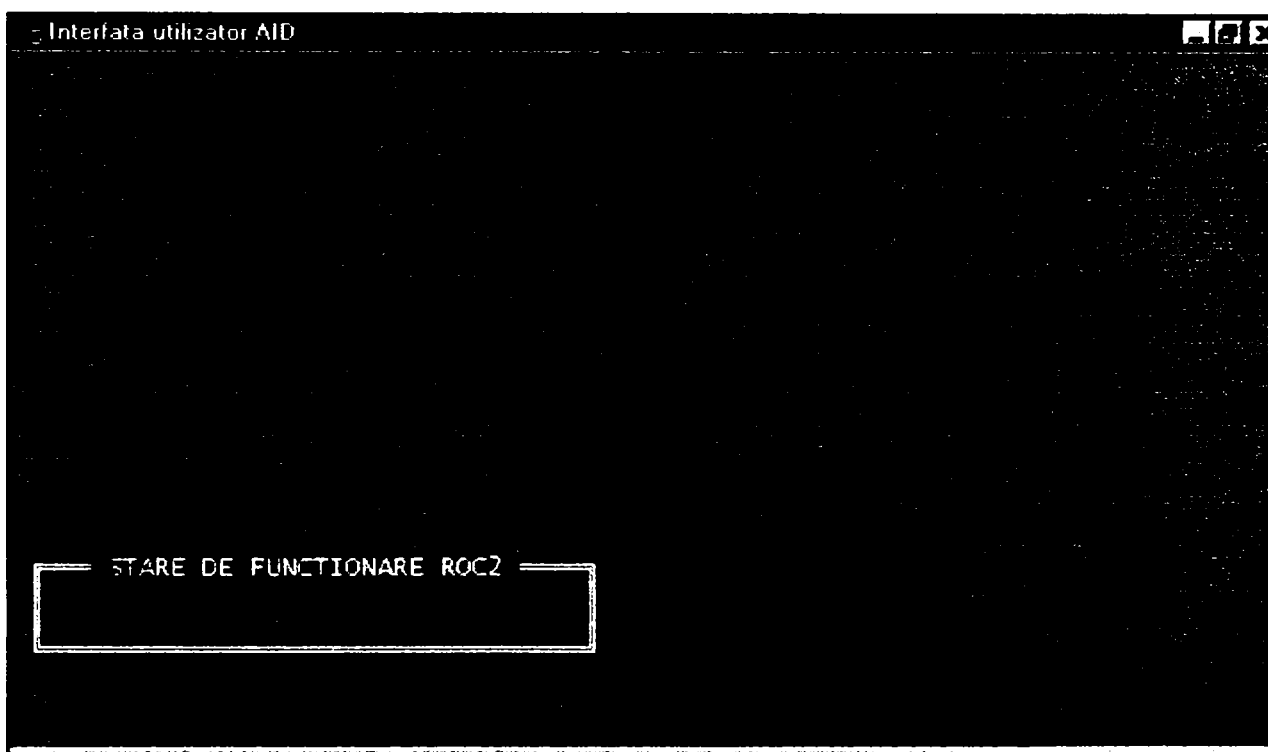


Fig.7.20. Fereastra "Stare de Funcționare ROC2"

Protocolul transmiterii comenzilor este foarte simplu. Calculatorul PC transmite comanda ("G") care este acceptată de controlerul AID prin emiterea înapoi a unui caracter "0". În caz de eroare este returnat după caz un caracter "1" sau "2", [2].

În programul de mai sus s-a implementat și *modul server* prin care se asigură recepția automată a cererilor de servicii, emise de controlerul de celulă și execuția comenzii aferente (lansare program, preluare stare de funcționare).

7.3.4. Conectarea informațională a calculatoarelor PC de interfață la controlerul de celulă

După conectarea echipamentelor de comandă (din cadrul celulei de fabricație), la câte un calculator PC de interfață, s-a trecut la realizarea conectării informaționale a acestora din urmă la controlerul de celulă. În acest scop fiecare calculator PC a fost dotat cu câte o **interfață LAN de tip Ethernet** (10Base2), iar interconectarea lor fizică s-a realizat prin intermediul unui cablu coaxial subțire de 50 ohmi, fig.7.21.

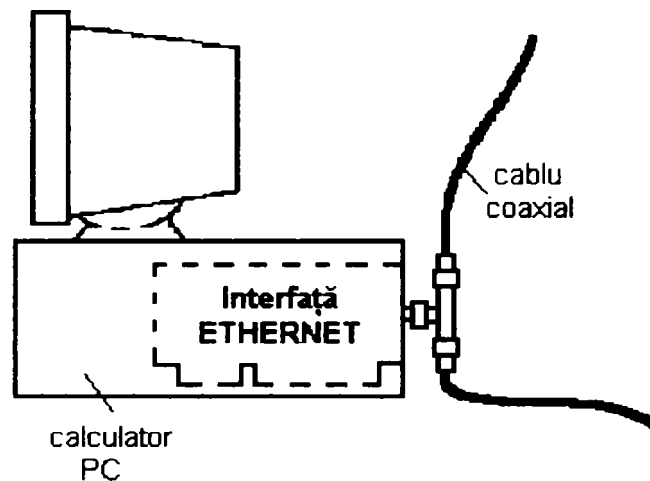


Fig.7.21. Fiecare calculator PC a fost dotat cu câte o interfață Ethernet

Pentru efectuarea schimburilor informaționale s-a adoptat o **arhitectură client-server**, în care pe rol de server se află echipamentele de fabricație din cadrul celulei iar pe rol de client se află controlerul de celulă.

Pornind de la această arhitectură client-server, s-a încercat dezvoltarea în cadrul celulei flexibile de fabricație, a unor structuri de comandă care au la bază un **model al stărilor generale de funcționare**, conceput în acest scop, fig.7.22., [15]. Practic s-a extins modelul stărilor de funcționare, adoptat în capitolul 5., și la celelalte stații ale celulei de fabricație.

Acest model, pe de o parte încearcă să stabilească stările de funcționare "generale" (adică uniform valabile pentru fiecare stație), iar pe de altă parte stabilește tranzițiile necesare, care cauzează trecerea dintr-o stare de funcționare în alta. Prin structurile de comandă elaborate, s-a încercat experimentarea unei **noi modalități de comandă a celulelor de fabricație**. Pe baza acestuia, controlerul de celulă își îndeplinește funcțiile de comandă prin:

- urmărirea stărilor curente de funcționare ale echipamentelor de fabricație din cadrul celulei,
- și la nevoie (în concordanță cu programul de fabricație), solicită efectuarea unor tranziții de stare.

Pentru realizarea acestor operații, controlerul de celulă utilizează o serie de cereri de servicii care sunt emise prin rețeaua LAN către echipamentele de fabricație, aflate pe rol de server.

În cazul celulei de fabricație realizată, controlerul de celulă nu poate accesa direct (prin rețeaua LAN), echipamentele de comandă existente (controlerele robot, echipamentul CNC600 și microcalculatorul Blue Earth 485). În consecință cererile de servicii prezentate sunt adresate calculatoarelor de interfață ale acestora. Aceste calculatoare, prin intermediul programelor instalate, au sarcina preluării cererilor de servicii adresate lor și realizarea tuturor operațiilor necesare efectuării tranziției solicitate, printr-o comandă adecvată a echipamentului de comandă din "subordine". În cazul cererii STATUS, calculatoarele emit un raport despre starea curentă de funcționare a echipamentului, în concordanță cu cele "detectate" prin legătură serială.

Practic aceste schimburi informaționale cu cereri de servicii, au loc între programul de comandă instalat pe controlerul de celulă (care este un program de tip Client) și programele instalate pe calculatoarele PC de interfață care, așa cum s-a menționat, oferă și un mod de lucru de tip "Server".

Pentru a fi mai ușoară adresarea acestor programe "Server", calculatoarele de interfață au primit denumiri consacrate, rezultate din denumirea stației sau a echipamentului de comandă, tab.7.5.

Componentele celulei de fabricație	Echipament de comandă	Denumire calculator de interfață (server)
Centru de prelucrare CP20UO	NUMERIK CNC600	CNC
Robot industrial RV-M1	CONTROLLER RV-M1	ROC1
Depozit tampon MR800	BLUE EARTH 485	DEP
Robot industrial AID	CONTROLLER AID	ROC2

Tab.7.5.Denumirile calculatoarelor de interfață aflate pe rol de server

Astfel de exemplu pentru trecerea echipamentului CNC600, din starea INACTIV (Idle) în starea ÎNCĂRCAT (Loaded) se poate utiliza cererea de serviciu:

load, CNC, %8

Așa cum se poate observa, prin această cerere este solicitată sub forma unei tranziții de stare, o operație de încărcare a programului %8 în memoria echipamentului CNC600, conform unei secvențe din programul de fabricație.

Programele de aplicație (cum este și %8), sunt stocate în controlerul de celulă. După acceptarea de către programul "Server" a cererii solicitate, programul %8 este încărcat întâi prin rețeaua LAN în calculatorul de interfață, care apoi încarcă (prin legătură serială) în memoria echipamentului CNC600, fig.7.23. După execuția cererii, "Server"-ul emite înapoi un raport nesolicitat asupra stării de funcționare.

Trebuie arătat că datorită modurilor specifice de funcționare ale echipamentelor de comandă menționate (stabilite de producător), nu s-a putut implementa toate cererile definite și prezentate. Excepție este cazul echipamentului CNC600.

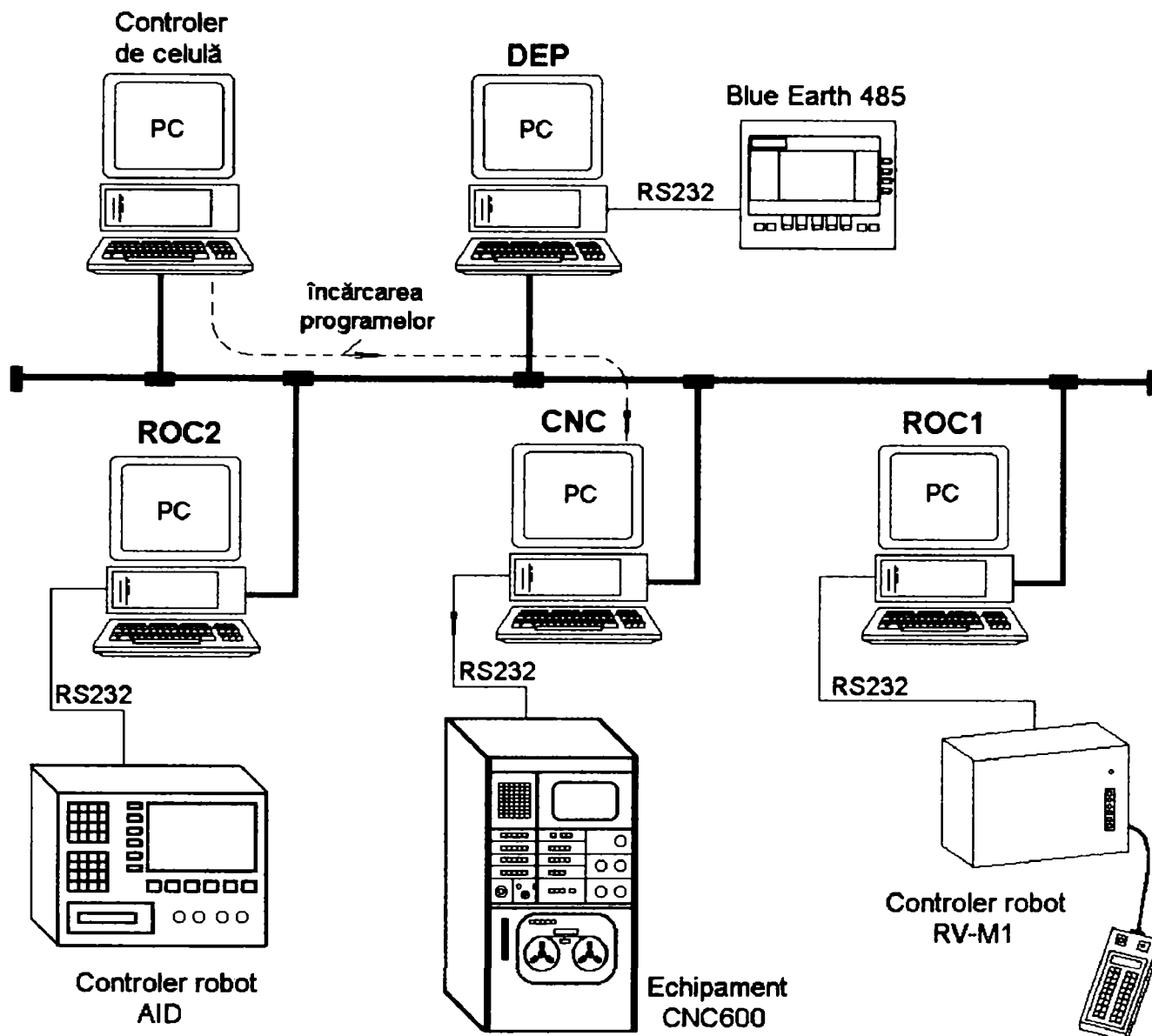


Fig.7.23. Ilustrarea încărcării programelor de la controlerul de celulă

Pentru experimentare, s-a adoptat un **proces de fabricație simplificat**, în care fiecare semifabricat, într-o succesiune prestabilită, este prelucrat pe rând, atât pe centrul de prelucrare cât și pe mașina GPR45NC.

Observație:

- Echipamentul de comandă numerică al mașinii GPR45NC n-a putut fi accesat de controlerul de celulă, printr-o legătură informațională, din motivele prezentate anterior. În consecință în cadrul experimentărilor s-a rezumat doar la alimentarea acestei mașini cu semifabricate, prin intermediul robotului AID.

În acest scop **s-au realizat programele de aplicație** și datele aferente, separat pentru fiecare echipament în parte, care apoi au fost introduse pe harddisk-ul controlerului de celulă. Excepție este cazul robotului AID, la care "programul" de aplicație (după instruire) a rămas stocat în memoria "nevolatilă" a controlerului robot și deci nu a fost încărcat, printr-o legătură informațională cu controler-ul de celulă, în timpul derulării procesului de fabricație. La celelalte echipamente însă, prima operație era ștergerea programelor existente în memorie ("prog"), după care sa realizat "încărcarea" programelor specificate, urmată de lansare în execuție.

Denumire "server"	Programe/ fișiere încărcate	Semnificație
CNC	F1	Fișier cu datele originii sistemului de coordonate utilizator ("origine piesă").
	%8	Program pentru preluarea inițială, de către masa mașinii, a unei poziții din care robotul RV-M1 poate accesa dispozitivul de fixare pentru realizarea operațiilor de încărcare/descărcare.
	%9	Program pentru prelucrarea semifabricatului pe centrul de prelucrare CP20UO.
ROC1	P1	Fișier cu datele situațiilor învățate aferente operațiilor de manipulare a semifabricatelor.
	R1	Program "task" pentru preluarea unui semifabricat din depozitul MR800 și introducerea respectiv fixarea acestuia în dispozitivul de pe masa centrului de prelucrare.
	R3	Program "task" pentru preluarea de pe centrul de prelucrare (din dispozitivul de fixare) a semifabricatului prelucrat, și așezarea acestuia într-un locaș al depozitului tampon MR800.
	R4	Program "task" pentru retragerea dispozitivului de prehensiune (din zona depozitului), la terminarea programului de fabricație (de exemplu după prelucrarea completă a celor 8 semifabricate depozitate).
DEP	M0	Program "task" pentru rotirea discului depozit dintr-o poziție "necunoscută" (de ex. la punerea sub tensiune), până în poziția de origine (locașul 1. către robotul RV-M1).
	M1	Program "task" pentru rotirea discului depozit în sens orar, cu o poziție față de cea curentă, (contorizarea pozițiilor este realizată de sistemul de comandă locală a depozitului MR800).
ROC2	-	

Tab.7.6. Programele și fișierele cu "date" încărcate de pe controlerul de celulă

Pentru experimentare s-a definit pentru controlerul de celulă un **mediu de programare aplicație**, care oferă un limbaj simplificat pentru realizarea programului de comandă a celulei de fabricație. Acest limbaj este asemănător limbajului BASIC, care pe lângă comenzile aferente emiterii cererilor de servicii, oferă în plus câteva instrucțiuni de control program de tipul:

- **salturi necondiționate** la o frază din program (GOTO, <număr frază>) indicat prin "număr";
- **salturi condiționate** de valoarea unei variabile asociate cu starea de funcționare a unui echipament (IF, <condiție>, GOTO, <număr frază>);
- **instrucțiuni de contorizare**, de exemplu a pieselor prelucrate (SET - inițializarea contorului specificat la o valoare dorită; INC – incrementarea valorii contorului).

Utilizând acest limbaj a rezultat următorul **program de comandă** a celulei de fabricație:

```
10 set,contorPiesa=0
15 load,CNC,F1
20 clear,ROC1,prog
25 load,ROC1,R1
30 load,CNC,%9
40 load,ROC1,R3
45 load,CNC,%8
50 load,ROC1,R4
55 select,CNC,%8
60 start,CNC
65 load,ROC1,P1
70 clear,DEP,prog
75 load,DEP,M0
80 start,DEP
85 clear,DEP,prog
90 load,DEP,M1
92 status,CNC
95 if,stareEchipament_CNC=READY,goto,135
97 goto,92
135 select,ROC1,R1
140 start,ROC1
145 start,ROC2
165 select,CNC,%9
170 status,ROC1
180 if,stareEchipament_ROC1=READY,goto,230
190 goto,170
230 start,CNC
300 select,ROC1,R3
310 status,CNC
320 if,stareEchipament_CNC=READY,goto,340
330 goto,310
340 start,ROC1
345 status,ROC2
350 if,stareEchipament_ROC2=READY,goto,370
355 goto,345
370 status,ROC1
380 if,stareEchipament_ROC1=READY,goto,400
390 goto,370
400 inc,contorPIESA
410 if,contorPIESA=8,goto,430
415 start,DEP
420 goto,135
430 select,ROC1,r4
432 start,ROC1
435 clear,CNC,%8
440 clear,CNC,%9
445 status,ROC1
450 if,stareEchipament_ROC1=READY,goto,470
460 goto,445
470 end
```

Trebuie arătat că în acest program de "test" sunt urmărite stările READY ale echipamentelor de fabricație, care indică faptul că execuția anterioară a programului de aplicație s-a terminat. De exemplu s-a terminat operația de prelucrare a semifabricatului pe centrul de prelucrare: `stareEchipament_CNC=READY` (fraza 320) și ca urmare se poate lansa programul "task" aferent operației de preluare a piesei de pe masa mașinii de către robotul RV-M1 (fraza 340). Bineînțeles pot fi urmărite și celelalte stări ale echipamentelor de fabricație.

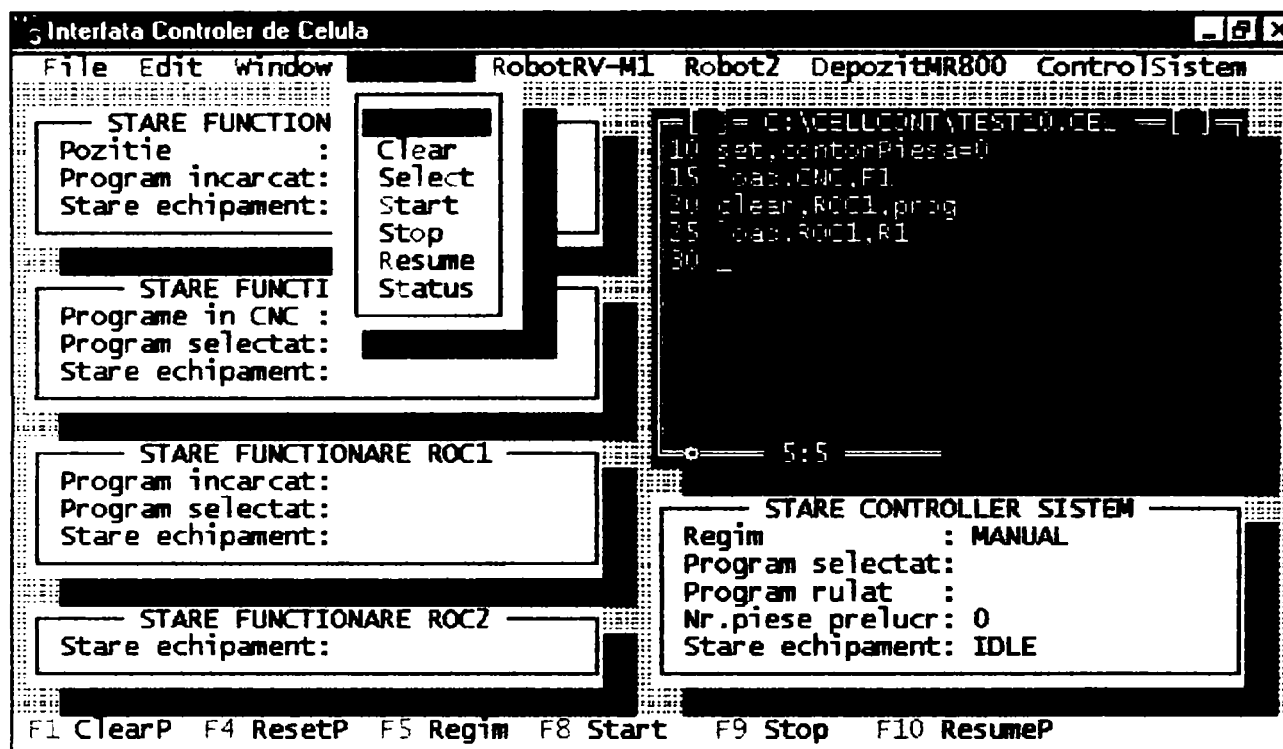


Fig.7.23. Interfața utilizator a programului Client realizat pentru controlerul de celulă

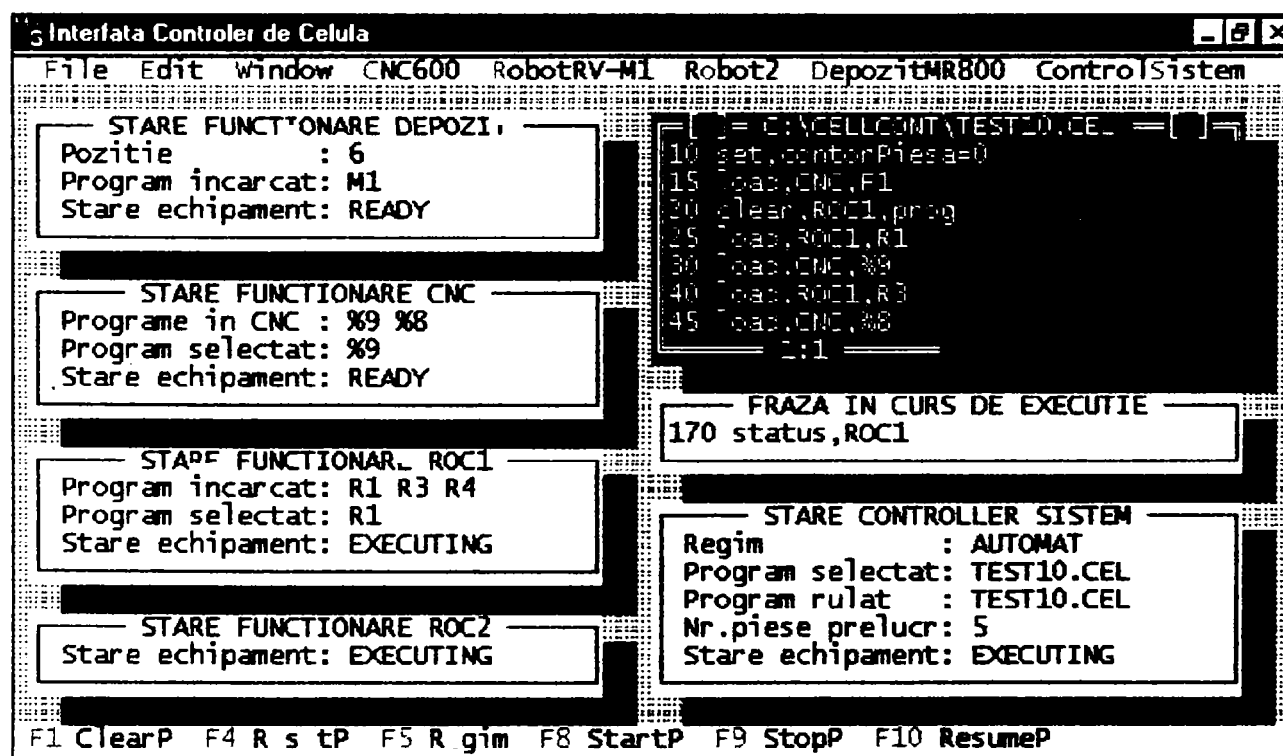


Fig.7.24. Interfața utilizator în timpul rulării programului de comandă "test10.cel"

Pentru rularea programelor de comandă de acest tip, respectiv pentru efectuarea comunicațiilor client-server aferente, s-a realizat un **program Client** care a fost instalat pe calculatorul PC aflat pe rol de controler de celulă, fig.7.23. Interfața utilizator a acestui program permite în **regim manual** experimentarea comenzilor aferente cererilor de servicii definite, și editarea programelor de comandă a celei.

În **regim semiautomat și automat** se poate selecta, lansa în execuție și rula programe de comandă realizate conform specificațiilor anterioare, fig.7.24. În timpul rulării unui program de comandă este afișată fraza care este în curs de execuție, respectiv un operator poate să urmărească stările de funcționare ale echipamentelor de fabricație, respectiv lista cu programele încărcate. De fapt stările de funcționare sunt afișate și în regim manual dar numai după ce operatorul transmite pe rând, la fiecare echipament, câte o cerere de serviciu STATUS.

Interfața permite de asemenea urmărirea stărilor de funcționare a “controlerului de sistem” (controler de celulă), regimul de lucru, programul rulat respectiv numărul pieselor prelucrate.

Observații:

1. Pentru transmiterea prin rețeaua LAN a cererilor de servicii precum și a rapoartelor de stare s-a utilizat aceeași structură deschisă oferită de Specificația Driverului de Pachete (Packet Driver Specification), care s-a dezvoltat pentru comunicația din mini-sistemul CIM2000 (paragraful 6.4.2).
2. Programul Client a fost prevăzut cu o listă a adreselor fizice Ethernet ale calculatoarelor de interfață. Un modul al programului se ocupă cu identificarea din această listă a adreselor Ethernet, aferente calculatoarelor PC pentru care în cadrul programului de comandă se utilizează doar “denumiri consacrate” (CNC, ROC1, DEP, ROC2).

7.4. Concluzii

În acest capitol, s-au prezentat unele experimentări efectuate, legate de integrarea fizică și informațională a unor module (roboți industriali, mașini unelte cu comandă numerică, depozit tampon) într-o celulă flexibilă de fabricație.

Față de cazul din mini-sistemul CIM2000, de aceasta dată erau accesibile specificațiile modurilor de conectare informațională, a echipamentelor de comandă aferente, la calculatoare PC de “interfață”. S-au utilizat de asemenea specificații deschise, pentru interconectarea calculatoarelor de interfață respectiv a controlerului de celulă printr-o rețea LAN de tip Ethernet.

S-a arătat că pentru această celulă de fabricație, a fost conceput și realizat un **sistem de comandă cu arhitectură client-server**. În această arhitectură, pe rol de client se află controlerul de celulă, iar pe rol de server se află calculatoarele PC de interfață împreună cu echipamentele de comandă la care sunt conectate serial.

S-a realizat **implementarea unor cereri de servicii** în cadrul programului Client, creat pentru controlerul de celulă și implementarea serviciilor aferente în programele Server, create pentru calculatoarele PC de interfață.

Având la bază cererile de servicii definite, s-a creat pentru experimentare un **mediu de programare aplicație**, care oferă pentru un utilizator realizarea unor programe de comandă (pentru celula de fabricație prezentată), într-un limbaj simplificat, asemănător limbajului BASIC.

Trebuie subliniat că prin experimentările efectuate, **nu s-a urmărit dezvoltarea unui nou limbaj de comandă** pentru celule de fabricație. Structurile de programare definite, au fost concepute numai ca suport pentru realizarea experimentărilor. S-a urmărit însă **dezvoltarea și experimentarea unor structuri de comandă**, care să asigure comanda celulei printr-o serie de cereri de servicii (comenzi) independente de echipament. În acest scop s-a conceput un model general al stărilor de funcționare, uniform valabilă pentru fiecare "stație" în parte.

În principiu cererile de servicii utilizate pot deveni "comenzi" comune pentru fiecare echipament de comandă, indiferent de tip (CNC, ROC sau eventual PLC) și producător. Experimentările efectuate au arătat că aceasta este posibilă. Exemplu în acest sens este **programul de comandă** elaborat și testat care utilizează cereri de servicii identice pentru comanda fiecărui echipament din cadrul celulei.

De fapt utilizarea unor **limbaje comune de comandă**, independente de echipament, este una dintre cerințele formulate pentru structurile deschise de comandă din cadrul Hipersistemelor CIM.

CAPITOLUL 8.

CONTRIBUȚII ORIGINALE. CONCLUZII FINALE

Contribuțiile originale, legate de dezvoltarea unor structuri deschise de automatizare a fabricației pentru Hipersisteme CIM robotizate, care au fost prezentate în capitolele anterioare sunt următoarele:

1. Sistematizarea problemei dezvoltării și introducerii în Hipersisteme CIM a structurilor deschise de automatizare a fabricației. Aceasta s-a concretizat prin:
 - analiza evoluției în cadrul Hipersistemelor CIM a structurilor deschise de comandă ("Open Control") respectiv prin
 - analiza tendințelor de dezvoltare a unor echipamente de comandă cu arhitectură deschisă ("Open Controller").
2. Elaborarea unor modele destinate structurilor deschise de comandă a roboților industriali din cadrul Hipersistemelor CIM. Scopul era dezvoltarea unor noi structuri de accesare de la distanță a resurselor fizice și logice ale roboților industriali, bazat pe concepția arhitecturii deschise și modulare a unui controler de robot. S-a realizat:
 - Elaborarea unui *model al resurselor fizice ale roboților industriali*. În cadrul acestui model s-au definit numai acele resurse fizice ale unui "sistem robot" care în general pot fi sau sunt necesare a fi accesate prin comandă de la distanță. S-a pus accent pe descrierea generală a resurselor fizice ale subsistemului manipulator respectiv ale subsistemului de conducere, propunând o nouă interpretare a acestora. De asemenea s-au stabilit suplimentar și principalele module (software) prin care se realizează controlul acestor resurse.
 - Elaborarea unui *model al resurselor logice ale roboților industriali*. În cadrul acestui model s-a încercat definirea acelor resurse care nu au corespondent fizic (hardware sau software) bine determinat, în schimb descriu stările generale de funcționare ale unui sistem robot. De asemenea, s-a propus o nouă diagramă pentru stările de funcționare ale roboților industriali, integrați în Hipersisteme CIM.
 - Elaborarea unui nou *model de accesare de la distanță a resurselor fizice și logice ale unui robot industrial*. Pe baza acestui model, accesul de la distanță la controlul resurselor se reduce la accesul standard la modulele software ale unui controler de robot cu arhitectură deschisă. În acest model au fost stabilite și funcțiile standard ale principalelor module, luând însă în considerare doar cerințele realizării comenzii de la distanță a robotului industrial. Meritul modelului constă în faptul că încearcă pentru prima oară tratarea problemei *comenzii deschise de la distanță* a unui robot industrial în paralel cu problema *arhitecturii deschise a controlerului de robot*.
3. Dezvoltarea unor structuri deschise de comandă pentru un mini-sistem CIM. Aceste structuri au fost destinate mini-sistemului CIM2000 de la Universitatea din Oradea. S-a realizat:
 - Elaborarea și implementarea parțială a unui proiect de dezvoltare a sistemului de comandă, prin care s-a prevăzut introducerea în mini-sistem a

unor structuri deschise de comandă. Ca rezultat al implementării proiectului, s-a prezentat noua structură a sistemului de comandă aferentă celulei de fabricație CF2101 din mini-sistemul CIM2000.

- Elaborarea unor structuri "deschise", pentru comanda de la distanță a robotului industrial RV-M1 din mini-sistemul CIM2000: *conectarea controlerului robot la un calculator PC; implementarea pentru robotul RV-M1 a modelului de accesare de la distanță a resurselor fizice și logice elaborat; implementarea diagramei cu stările de funcționare.*
 - Elaborarea unor structuri deschise MMS, pentru semaforizarea accesului la resursele comune, din cadrul mini-sistemului CIM2000. Această structură a fost utilizată pentru semaforizarea accesului la comanda singurului robot industrial din mini-sistem, considerat a fi o resursă comună. Însă ea poate fi extinsă și la celelalte stații, cu condiția implementării și în cadrul acestora, a unor structuri deschise de comandă.
4. Construirea unei celule flexibile de fabricație integrată cu calculatorul. Scopul a fost dezvoltarea și experimentarea unor structuri deschise de comandă, care să asigure comanda celulei printr-o serie de comenzi "standarde", independente de echipament. S-a realizat:
- Conceperea unei celule flexibile de fabricație și integrarea fizică în aceasta, a unor module de fabricație adecvate (roboți industriali, mașini-unelte cu comandă numerică, depozit tampon), existente la Universitatea Oradea.
 - Conceperea și realizarea sistemului de comandă a celulei de fabricație prin conectarea informațională a echipamentelor de comandă, la calculatoarele PC de "interfață" și conectarea acestora din urmă la "controlerul de celulă". De asemenea a fost conceput și realizat un sistem de comandă locală pentru depozitul tampon.
 - Conceperea și implementarea unui model general al stărilor de funcționare, uniform valabil pentru fiecare "stație" în parte.
 - Conceperea și realizarea unei arhitecturi de comandă de tip client-server. S-a concretizat în crearea unor programe "Server" instalate pe calculatoarele PC de interfață și a unui program "Client" instalat pe controlerul de celulă.
 - Conceperea și realizarea unui mediu de programare aplicație. Acesta poate fi utilizat pentru crearea programelor de comandă pentru celula de fabricație realizată.

Așa cum rezultă și din "lista" contribuțiilor originale, s-a reușit dezvoltarea unui set minim de structuri "deschise" de comandă, care însă oricând pot fi utilizate pentru integrarea informațională sau pentru comanda unei celule flexibile de fabricație. Prin intermediul experimentărilor efectuate s-au pus bazele unor structuri de comandă independente de echipament, ceea ce este o cerință hotărâtoare la dezvoltarea structurilor deschise, destinate Hipersistemelor CIM.

Printre direcțiile posibile ale continuării cercetărilor se poate aminti: dezvoltarea unor structuri deschise de comandă, pe baza suitei de protocoale TCP/IP (specifice "Internetului") sau pe baza unei variante industriale a protocolului Ethernet numit: EtherNet/IP.

BIBLIOGRAFIE

- [1]. Amaranth, P. A., *MMS: An Application Protocol*, IN: CIM Technology, May 1987
- [2]. Assistance Industrielle Douphinoise: *Robot AID V5 – Documentație tehnică*, Grenoble, Meylan, 1986
- [3]. Barabás, T., *Asupra structurii informaționale a mini-sistemului CIM2000*, 2nd International Conference on Renewable Sources and Environmental Electro-Technologies, Analele Universității din Oradea, Fascicula Electrotehnică, 1998
- [4]. Barabás, T., *Asupra posibilităților de implementare a Protocolului de Automatizare a Fabricației în cadrul mini-sistemului CIM2000*, 2nd International Conference on Renewable Sources and Environmental Electro-Technologies, Analele Universității din Oradea, Fascicula Electrotehnică, 1998
- [5]. Barabás, T., *Asupra integrării unor ministații de lucru într-un sistem flexibil educațional prin intermediul sistemului micro-robot RV-M1*, Analele Universității din Oradea, Fascicula Mecanică, 1995
- [6]. Barabás, T., *Stadiul actual al cercetărilor și realizărilor în domeniul structurilor MAP pentru hipersisteme robotizate CIM*, Referat nr.1, 1998
- [7]. Barabás, T., *Programe și interfețe specifice pentru structuri MAP incluse în hipersisteme robotizate CIM*, Referat nr.2, 1999
- [8]. Barabás, T., *Integrarea informațională a roboților industriali în Hipersisteme CIM*, Referat nr.3, 2000
- [9]. Barabás, T., *On the Implementation of the MMS Robot Operation States for RV-M1 Robot System*, 3rd International Conference on Renewable Sources and Environmental Electro-Technologies, Analele Universității din Oradea, Fascicula Electrotehnică, 2000
- [10]. Barabás, T., *On the Implementation of the Token Semaphore Management in CIM2000 Minisystem*, 3rd International Conference on Renewable Sources and Environmental Electro-Technologies, Analele Universității din Oradea, Fascicula Electrotehnică, 2000
- [11]. Barabás, T., *Asupra posibilităților de dezvoltare a mini-sistemului educațional de fabricație CIM2000*, Simpozion Național de Robotică, Băile Felix, 2000
- [12]. Barabás, T., *Asupra posibilităților de modernizare a structurii informaționale mini-sistemului CIM2000*, Simpozion Național de Robotică, Băile Felix, 2000
- [13]. Barabás, T., *Ethernet/IP an Open Standard for Real Time Control of Industrial Applications*, 4th International Conference on Renewable Sources and Environmental Electro-Technologies, Analele Universității din Oradea, Fascicula Electrotehnică, 2002
- [14]. Barabás, T., *Open Architecture Robot Controllers*, 4th International Conference on Renewable Sources and Environmental Electro-Technologies, Analele Universității din Oradea, Fascicula Electrotehnică, 2002
- [15]. Barabás, T., *Implementarea și monitorizarea stărilor de funcționare standard în sisteme CIM*, Conferința Națională de Robotică, Craiova, 17-19 octombrie 2002
- [16]. Blue Earth Research Inc.: *Blue Earth MICRO-485 – USER'S MANUAL*, Blue Earth Research, Mankato, Minnesota USA, 1993
- [17]. BOEING: *D38550-6. Communication Requierements to Boeing Network*, 1998 (www.boeing.com/companyoffices/doingbiz/edaas/pdf/D38550-6_Comm.pdf)
- [18]. Bulăceanu, C., *Rețele de calculatoare - Arhitecturi prezente și viitoare*, Editura Tehnică, București, 1995
- [19]. Castori, P., Pleinevaux, P.: *MMS Product list*, 1996 (http://icawww.epfl.ch/mms/mms_MMSproducts.txt)
- [20]. Ceapăru, M., *Comunicația prin intermediul rețelelor de calculatoare*, Editura Tehnică, București, 1996

- [21]. Chrysler, Ford Motor and General Motors: *Requirements of Open, Modular Architecture Controllers for Applications in the Automotive Industry*, 1994, White Paper - Ver. 1.1. (<http://www.arcweb.com/omac>)
- [22]. Crowder, R., *The MAP Specification*, In: Control Engineering, October 1985
- [23]. Csurgai, G., Haidegger, G., *Gyártásautomatizálási informatikai protokoll*, MTA SZTAKI, Budapest, 1986
- [24]. Degem System Ltd: *CIM-2000 Mechatronics Training System*, 1992
- [25]. Degen, J., *Open Architecture Controls for the Robotics Industry*, In: Proceedings of "Open architecture Control for the Robotics Industry " workshop, Orlando, Florida, February 9-10, 2000
- [26]. Dinsmor, C., *FANUC Open Architecture*, IN: Proceedings of "Near-Term Objectives Workshop - Open Architecture Control in Robotics, Rochester Hills, January 29, 2001
- [27]. Dubuisson, O., *ASN.1.- Communication between heterogeneous systems*, Open System Solutions, Nokalva, 2000 (<http://www.oss.com>)
- [28]. Dwyer, J., Ioannou, A., *MAP and TOP*, Halsted Press, John Wiley and Sons, Inc., New York, 1988
- [29]. EMUG European MAP/TOP Users Group: *Manufacturing Automation Protocol Specification, version 3.0.*, Vol.I-V, 1989
- [30]. EMUG European MAP/TOP Users Group: *MAP/TOP Product & Directory for Manufacturing Enterprises*, United Kingdom, 1991
- [31]. EMUG European MAP/TOP Users Group *MAP/MMS Product & Installation Summary*, United Kingdom, 1994
- [32]. ESPRIT III. Open System Architecture Controls within Automation Systems – OSACA I & II Final Report, (<http://www.osaca.org/>), April 1996
- [33]. ESPRIT III. Open System Architecture Controls within Automation Systems – OSACA Handbook, (<http://www.osaca.org/>), April 1996
- [34]. Falk, H., Tomlinson, J., *Manufacturing Message Specification (MMS)*, In: MAP/TOP Users Group Summary, January 1987
- [35]. FANUC Robotics N. A.: *Wave I Status and Wave II & III Directions*, 5th RIA/NIST Workshop on Open Architecture Control in Robotics, Rochester Hills, 9 January 2002
- [36]. Farnady, L., *Számítógépes hálózatok*, Széchenyi István Főiskola, Győr, 1995
- [37]. Fiedler, P., Schilb, C., *Open Architecture robot controllers and workcell integration*, FABTECT 1997 Conference Proceedings, Chicago, IL, 1997
- [38]. Fiedler, P., Dehof, K., *Workcell Communications: Connectivity, Man-Machine Interfaces, and Multi-System Management*, 1997 (<http://www.roboticonline.org/> Technical Papers)
- [39]. FTP Software, Inc.: *PC/TCP - Packet Driver Specification, rev.1.09.*, 1989 (http://www.crynwr.com/packet_driver.html)
- [40]. GE Fanuc: *CNC & Laser*, 2001, (<http://www.gefanuceur.com/3/31.asp>)
- [41]. Gravel, D., *Open Architecture Controllers Extensibility Issues*, In: Proceedings of "Near-Term Objectives Workshop – Open architecture Control for Robotics", Ypsilanti, Michigan, June 28, 2000
- [42]. Gressier-Soudan, E., Lefebvre, M., Natkin, St., *TCP/IP manufacturing applications: an experiment with MMS over RPC*, ULPAA Web Conference, 1995 (<http://cedric.cnam.fr/personne/gressier/ULPAA95/ULPAA95.html>)
- [43]. Haidegger, G., *Open Systems in Manufacturing European and World-wide Harmonized Initiatives*, Automation '95 Conference with International Participation, Budapest, September 5-7, 1995
- [44]. Haidegger, G., *Trends in Industrial Data Networks*, MicroCAD 2000 – Nemzetközi Számítástechnikai Tudományos Konferencia, Miskolc, 23-24 febr., 2000
- [45]. Haidegger, G., Kuba, R., *Highly Automated Flexible Assembly System*, Automation '95 Conference with International Participation, Budapest, September 5-7, 1995
- [46]. Haidegger, G., Nacsá, J., *Shop-Floor Communication with OSACA-Compliant Controllers*, IEEE International Workshop on Factory Communication Systems, Barcelona, October 1-3, 1997

- [47]. Hornbeck, D., *EtherNet IP: An Open Standard for Real-Time Control Over Ethernet*, In: Proceedings of "Near-Term Objectives Workshop – Open architecture Control for Robotics", Ypsilanti, michigan, June 28, 2000
- [48]. Heilala, J., *Open Real-time Robotics Control - PC Hardware, Windows/VxWorks Operating Systems and Communication*, 2001 (www.automationit.hut.fi)
- [49]. IEEE802.3: *Information Technology - Telecommunication & Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, 2002
- [50]. Interbus Club: *INTERBUS BASICS*, 2000, (www.interbusclub.com)
- [51]. ISO/IEC 7498-1: *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*, 1994
- [52]. ISO 8571: *Information processing systems – Open Systems Interconnection – File Transfer, Access and Management*, 1988
- [53]. ISO/IEC 8649: *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element*, 1996
- [54]. ISO/IEC 8650: *Information technology – Open Systems Interconnection – Connection-oriented protocol for the Association Control Service Element: Protocol specification*, 1996
- [55]. ISO/IEC 8802-2: *Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 2: Logical Link Control*, 1998
- [56]. ISO/IEC 9040: *Information technology – Open Systems Interconnection – Virtual Terminal Basic Class Service*, 1997
- [57]. ISO/IEC 9041: *Information technology – Open Systems Interconnection – Virtual Terminal Basic Class Protocol*, 1997
- [58]. ISO/IEC 9506-1: *Industrial Automation Systems - Manufacturing Message Specification - Part 1: Service Definition*, 2000
- [59]. ISO/IEC 9506-2: *Industrial Automation Systems - Manufacturing Message Specification - Part 2: Protocol Specification*, 2000
- [60]. ISO/IEC 9579: *Information technology – Remote database access for SQL with security enhancement*, 2000
- [61]. ISO/IEC 9594: *Information technology – Open Systems Interconnection – The Directory*, 1988
- [62]. ISO 9787: *Manipulating Industrial Robots – Coordinate Systems and Motion*, 1990
- [63]. ISO 11898: *Road vehicles - Interchange of digital information – Controller area network (CAN) for high-speed communication*, 1993
- [64]. Ivănescu, M., *Roboți industriali*, Editura "Universitaria", Craiova, 1994
- [65]. Katzel, J., *Moving Down the Path to Open Systems*, In: Plant Engineering, 1997
- [66]. Klein, M., Balph, T., *Carrierband is Low-cost, Single-channel solution for MAP*, In: Computer Design, February 1, 1986
- [67]. Kneifel, B., *Open Architecture: Where should we spend our efforts?*, In: Proceedings of "Near-Term Objectives Workshop – Open architecture Control for Robotics", Ypsilanti, michigan, June 28, 2000
- [68]. Knipp, R., *Broadband Technology*, In: OSI MAP TOP Educators & Consultants, 1987
- [69]. Kovács, Fr., Túsz, Fr., Varga, Șt., *Fabrica viitorului. Introducere în producție*, Editura Multimedia International, Arad, 1999
- [70]. Kovács, Fr., Varga, Șt., Pau, V.C., *Introducere în robotică*, Editura Printech, București, 2000
- [71]. Kovács, Fr., Grigorescu, S., Rădulescu, C., *Sisteme de fabricație flexibilă robotizate*, Universitatea Tehnică Timișoara, 1994
- [72]. Kovács, Gy., Haidegger, G., *Open Systems in Manufacturing*, Second International Workshop on Learning in Intelligent Manufacturing Systems, Budapest, April 20-21, 1995

- [73]. Kovács, Gy., Haidegger, *MAP Networking to assist CIM Implementations*, Computer-Aided Technologies for Management & Productivity International Conference and Trade Show, Budapest, Sept. 25-27., 1991
- [74]. KUKA Roboter GmbH, <http://www.kuka-roboter.de/english/index.html>
- [75]. Langmann, R., *Interbus Basics*, University of Applied Sciences, Düsseldorf, 2000
- [76]. Lamouth, J., *ASN.1 Complete*, Open System Solutions, Nokalva, 1999 (<http://www.oss.com>)
- [77]. Lantos, B., *Robotok irányítása*, Akadémiai Kiadó, Budapest, 1991
- [78]. Manufacturing Data Systems: *OpenCNC Architecture, OpenCNC Application Programming Interface*, 2000, (<http://www.mdsi2.com>)
- [79]. Matias, R., *Open Architecture Controls for Robots*, In: Proceedings of "Open architecture Control for the Robotics Industry " workshop, Orlando, Florida, february 9-10, 2000
- [80]. Matias, R., *OMAC API Workgroup Status Report*, In: Proceedings of "Open architecture Control for the Robotics Industry " workshop, Orlando, Florida, February 9-10, 2000
- [81]. Motion Engineering: XMP Series Product Overview, 1998, <http://www.motioneng.com>
- [82]. MSTC/JOP: *FL-net Protocol Specifications Ver1.0*, July, 1996, (<http://www.mstc.or.jp/faop/misc/spec-e.html>)
- [83]. Munz, H., *LP- VxWin VxWorks Together with Windows on the same PC*, Real-Time Magazine 1997-2. (http://www.dedicated-systems.com/magazine/97q2/1997q2_p047.pdf)
- [84]. Nacsá, J., *Comparison of three different open architecture controllers*, Proc. of IFAC MIM, Prague, 2-4 Aug. 2001, pp. 134-138
- [85]. Mitsubishi Corp.: *Industrial Micro-Robot System RV-M1*, Manual de operare, 1992
- [86]. Moore, G., *Manufacturing Automation Protocol: Mapping the Factory of the Future*, in: Electronics and Power, Vol.32, No.4, April 1986
- [87]. Nagy, G., Haidegger, G., *MMS-MAP Based and Object Oriented Software Approach for Controlling Shop-Floor Devices in a CIM Environment*, 9th International Conference on Control Systems and Computer Science, Bucharest, 25-28 May 1993
- [88]. Nokleby, C., *Technical report: Object Oriented MMS Analysis, and design Method for Specification of Remote Controlled Application Processes*, Danfoss AIS, Denmark, Marts 1991
- [89]. OMAC USERS GROUP: *OMAC Basline Architecture - Functional Requirements*, February 2002, (<http://www.omac.org/>)
- [90]. Open DeviceNet Vendor Association Inc. (ODVA): *What is the DeviceNet? – Technical Overview*, , 2001 (www.odva.org/10_2/05_tech/WhatisDeviceNet.pdf)
- [91]. Open I. T. Corp: *Mini-MAP '93, A Practical Application of Standards to Manufacturing Cell Networks*, Rochester, 1994
- [92]. OSE Consortium, OSEC: *Open System Env. for Controller Architecture Draft*, 1996, (<http://www.mli.co.jp/OSE/>)
- [93]. Parr, E. A., *Programable Controllers: An Engineer's Guide*, Butterworth-Heinemann Ltd, Oxford, 1993
- [94]. Petterson, G., *MMS – The Universal Tool for CIM Applications*, Motorola Computer Systems, Cedex, France, 1993
- [95]. Pilat, Fl. ș.a., *Introducere în INTERNET*, Editura Teora, București, 1995
- [96]. Ranky, P.G., *Computer Networks for World Class CIM Systems*, Cimware Limited, Guildford, Surrey, England, 1990
- [97]. Ranky, P.G., *Flexible Manufacturing Cells and Sytems in CIM*, CIMware Limited, Guildford, Surrey, England, 1990
- [98]. Rădulescu, C., *Robocare și sisteme de robocare*, Editura MIRTON, Timișoara, 2000
- [99]. Rutlege, G., *Open Architecture Interfaces to Robot Controllers*, In: Proceedings of "Near-Term Objectives Workshop – Open architecture Control for Robotics", Ypsilanti, michigan, June 28, 2000
- [100]. Samson AG: *FOUNDATION Fieldbus – Technical Information*, 2000, (<http://www.samson.de>)

- [101]. Scheer, A. W., *CIM – Computer Steered Industry*, Springer-Verlag, Berlin-Heidelberg-New York-London-Paris-Tokyo, 1988
- [102]. SISCO Inc.: *MMS – EASE*, Reference Manual, SISCO Inc., 1997
- [103]. SISCO, Inc.: *Overview and Introduction to the Manufacturing Message Specification (MMS)*, 1995, (<http://www.sisconet.com>)
- [104]. SISCO products: *MMS-EASE*, 2000, (<http://www.sisconet.com/mmsease.htm>)
- [105]. Somogyi, A., *Integrating Control and Information in Manufacturing Systems using Open Protocols*, IN: Proceedings of "Near-Term Objectives Workshop - Open Architecture Control in Robotics, Rochester Hills, January 29, 2001
- [106]. Valenziano, A., Demartini, C., *MAP and TOP Communications – Standards and Applications*, Addison-Wesley Publishing Company Inc., 1992
- [107]. Verissimo, P., Melro, S., Casimiro, A., *Distributed Industrial Information Systems: Design and Experience*, Proceedings of the 2nd International Conference on Architectures and Design Methods for Balanced Automation Systems (BASYS'96), Lisboa, Portugal, June 1996
- [108]. Vukobratovic, M., *Applied Dynamics of Manipulation Robots*, Springer-Verlag, Berlin, Heidelberg, New York, 1989
- [109]. Wells, J., Hamor, J., *Robot Support for Ethernet/IP at General Motors*, 5th RIA/NIST Workshop on Open Architecture Control in Robotics, Rochester Hills, 9 January 2002
- [110]. Wells, J., Heaton, J., *Wave 1 Networking Standards: Features and Functionality*, IN: Proceedings of "Near-Term Objectives Workshop - Open Architecture Control in Robotics, Rochester Hills, January 29, 2001
- [111]. World Federation of MAP/TOP Users Groups: *European MAP Installations Summary*, San Francisco, USA, 1993
- [112]. ...*NUMERIK CNC600 – Documentație tehnică*, 1986