

Conducerea fabricației discrete, repetitive dirijate de produs

1. Introducere

Principala preocupare a producătorilor a fost dintotdeauna îndeplinirea cerințelor pieței și, în mod evident, să poată face față rapid schimbărilor impuse de piață. Inițial, principalul criteriu de alegere a produselor pentru client era prețul. Mai apoi, calitatea produsului și timpul de livrare au devenit un indice important al succesului unui producător.

În ultimele decenii competiția între producători a devenit tot mai strânsă, iar fidelitatea consumatorilor pentru un anumit produs sau producător a scăzut. O soluție adoptată de companii este producția produselor personalizate, adică fabricarea produselor în volume mixte, la prețuri mici și calitate bună. Astfel se face trecerea la o piață centrată pe client în care se pune accentul pe inovare, personalizare și oferirea de servicii la un preț cât mai mic.

Un producător agil este acela care reușește să păstreze un echilibru între rapiditatea lansării produselor pe piață, menținerea unui cost total mic și creșterea gradului de satisfacție a clientului, cu alte cuvinte este capabil să răspundă rapid la schimbările de pe piață.

Companiile caută să micșoreze ciclul de viață al produsului, timpul de lansare pe piață, timpul de livrare și să crească varietatea produselor, în timp ce mențin calitatea produselor și costurile de investiții constante (Bussmann and McFarlane, 1999).

Schimbările din piață se reflectă în mod evident și asupra domeniului industriei. Astfel, tendința actuală în acest domeniu este de a se face trecerea de la producția în masă la producția produselor individuale, personalizate. Sistemele de fabricație trebuie să fie flexibile pentru a putea face față schimbării constante a produselor și a creșterii complexității procesului de fabricație, precum și a menținerii unui preț cât mai mic.

Cerințele pentru sistemele de fabricație au evoluat de la criterii de performanță tradiționale, ca optimalitatea producției la cerințe de reactivitate, adaptabilitate și vizibilitate (Trentesaux and Thomas, 2012). Pentru a îndeplini aceste criterii, arhitectura de conducere a sistemelor de fabricație a evoluat de la structura tradițională, centralizată la cea descentralizată în care funcția de control este distribuită către mai multe entități din sistem. Aceste entități sunt autonome, au capacități decizionale și de cooperare și comunicare proprii.

O altă tendință din domeniul sistemelor de fabricație, care poate fi considerată complementară celei pentru utilizarea sistemelor distribuite este de a face trecerea de la produse pasive la produse active, adică de a adăuga inteligență produselor. Astfel, informațiile legate de starea produsului sunt legate în permanentă de produsul fizic.

Se consideră că utilizarea conceptului de produs inteligent în sistemele de fabricație îmbunătățește ciclul de viață al produsului prin creșterea vizibilității și trasabilității acestuia și prin creșterea accesibilității informației de-a lungul lanțului de aprovizionare. Un exemplu concret în care implementarea unui astfel de concept aduce beneficii este situația mai multor companii partenere, între care trebuie să existe cooperare și acces facil către informații.

Deasemenea, produsul inteligent poate duce la îmbunătățirea relației cu consumatorul, oferindu-i acestuia posibilitatea de a personaliza produsele comandate și de a controla propriile comenzi, spre exemplu clientul poate modifica date privind livrarea comenzii.

Conceptele de sisteme de fabricație holonice, produs inteligent, sisteme de conducere centrate pe produs, sisteme de fabricație inteligente, sisteme de fabricație bazate pe sisteme multi-agent sunt câteva dintre conceptele care se regăsesc în atenția cercetătorilor din domeniul fabricației și definesc noile tendințe în proiectarea sisteme de fabricație.

2. Sisteme distribuite de conducere a proceselor de fabricație flexibile

2.1. Descrierea sistemelor de fabricație flexibile și a tendințelor actuale în conducerea sistemelor flexibile de fabricație

Sistemele flexibile de fabricație sunt sisteme de fabricație care sunt capabile de a reacționa în cazul apariției unor schimbări previzibile sau imprevizibile. Există două niveluri de flexibilitate: flexibilitatea resurselor (de ex. mașini, roboți) și flexibilitatea fluxului materialelor. În primul caz, sistemul poate procesa tipuri noi de produse, iar ordinea în care se execută operațiile pentru un anumit produs este variabilă. În cel de-al doilea caz, echipamentele din celulă de fabricație pot procesa mai multe operații, iar ruta produselor procesate prin celulă de fabricație poate să fie diferită.

Beneficiile aduse de sistemele flexibile de fabricație sunt: costuri mai mici la schimbarea tipului de produse fabricate, toleranță la defect - în cazul defectării unei resurse, o altă resursă poate executa aceleași operații și timp total de procesare (*lead time*) a unui produs mai mic.

Dezavantajele sistemelor flexibile de fabricație sunt date de complexitatea configurării resurselor pentru executarea diferitelor operații și costurile mari pentru achiziția echipamentelor complexe.

Există mai multe tipuri de sisteme de fabricație, în funcție de modurile de producție: flow shop, job shop și open shop.

În cadrul sistemelor de tip producție continuă (flow-shop) posturile de lucru sunt dispuse în linie, fiecare produs urmând aceeași secvență de operații și aceeași ruta prin celulă de fabricație. Resursele într-un sistem de tip flow-shop sunt dedicate execuției unei anumite operații. Acest tip de sistem este folosit în cazul volumelor mari de producție și a unei cereri stabile.

Cel mai complex tip de sistem de fabricație este cel de tip discret (discontinuu)/job-shop în care operațiile sunt ordonate, însă posturile de lucru sunt dispuse în ordine aleatoare în celulă de fabricație. Acest tip de sistem permite fabricarea în volume mixte, personalizarea produselor și prioritizarea comenzilor. Resursele pot executa diferite operații, iar ruta produselor prin celulă de fabricație este flexibilă. Deasemenea, nu există nicio constrângere pentru rutarea produsului sau pentru numărul de operații executate la un post de lucru.

Sistemele de tip job-shop sunt caracterizate de flexibilitate, deoarece pot fi adăugate sau înlocuite resurse fără a fi oprită producția.

Al treilea tip de sistem de fabricație, open-shop este similar sistemelor job-shop, însă nu există nicio constrângere a ordinii în care se execută operațiile.

În cazul ultimelor două tipuri de sisteme este necesară implementarea unor algoritmi pentru planificarea producției și alocarea resurselor. În general, problemele sunt formulate astfel: se consideră n joburi, m posturi de lucru și timpul de procesare pentru fiecare job la fiecare post de lucru. Un job poate fi procesat la un post de lucru la un anumit moment, iar fiecare post de lucru poate procesa un singur job la un anumit moment. Scopul algoritmului este de a minimiza timpul total de procesare a tuturor joburilor. Ordinea în care sunt procesate joburile pentru un anumit produs sunt variabile pentru sistemele de tip open-shop.

Un sistem de conducere trebuie să îndeplinească următoarele cerințe: fiabilitate, toleranță la defect, robustețe, scalabilitate, flexibilitate, reconfigurabilitate și adaptabilitate. Flexibilitatea este definită ca posibilitatea de a modifica sau extinde arhitectura de control. Arhitectura unui sistem de conducere stabilește în ce măsură sistemul de conducere poate fi modificat în viitor pentru a respecta cerințele de business.

În (Bussmann and McFarlane, 1999) autorii precizează că sistemul de conducere trebuie să fie reactiv și pro-activ, adică sistemul trebuie să recunoască situațiile critice, să ia decizii cu privire la situația dată și să execute acțiunile respective.

Tendința actuală în proiectarea sistemelor de conducere a fabricației este trecerea de la dezvoltarea unor sisteme de conducere care să ofere optimalitate în condiții deterministe la sisteme reconfigurabile, orientate către servicii, care să fie stabile în medii stocastice.

2.2. Analiza tipurilor de arhitecturi de conducere

Rolul sistemelor de conducere a fabricației prin servicii interne este de a planifica și coordona activitățile de producție care transformă materiile prime în produse finite, mai exact sistemul de conducere controlează planificarea și rutarea produselor și alocarea resurselor din celula de fabricație. Obiectivul unui sistem de conducere este definit adesea sub forma unei valori sau criteriu ce trebuie optimizat, iar în proiectarea sistemului de conducere se definește strategia pe care sistemul o va folosi pentru a atinge acel obiectiv.

Eficacitatea unui sistem de conducere este dată de rapiditatea cu care sistemul detectează schimbările intervenite în sistem și acționează pentru a elimina perturbatiile.

Arhitectura de conducere trebuie să descrie comportamentul global al sistemului de fabricație și trebuie aleasă în funcție de cerințele de producție care reflectă obiectivele de business propuse. Din acest motiv sistemele de conducere au un rol critic în succesul oricărei întreprinderi.

În (Dilts et al, 1991) este prezentată evoluția sistemelor de conducere, de la sistemele tradiționale la cele descentralizate și sunt definite 4 tipuri de arhitecturi: centralizată, ierarhică, ierarhică-modificată și heterarhica. Fig. 2.1 prezintă cele 4 tipuri de arhitecturi de conducere.

Sistemele de conducere tradiționale sunt sistemele care au o arhitectură centralizată. Specific acestui tip de arhitectură este elementul decizional centralizat care îndeplinește funcțiile de planificare și procesare a informațiilor din sistem.

Deciziile se transmit de sus în jos (*top-down*) de la elementul central la nivelul operațional care este constituit din entități non-inteligente ce au rolul de a executa comenzile

primite de la nivelul superior. Informațiile despre starea resurselor din celulă de fabricație se transmis de jos în sus (bottom-up) către elementul centralizat. Aceste informații sunt folosite de către unitatea de control pentru luarea deciziilor la nivel global.

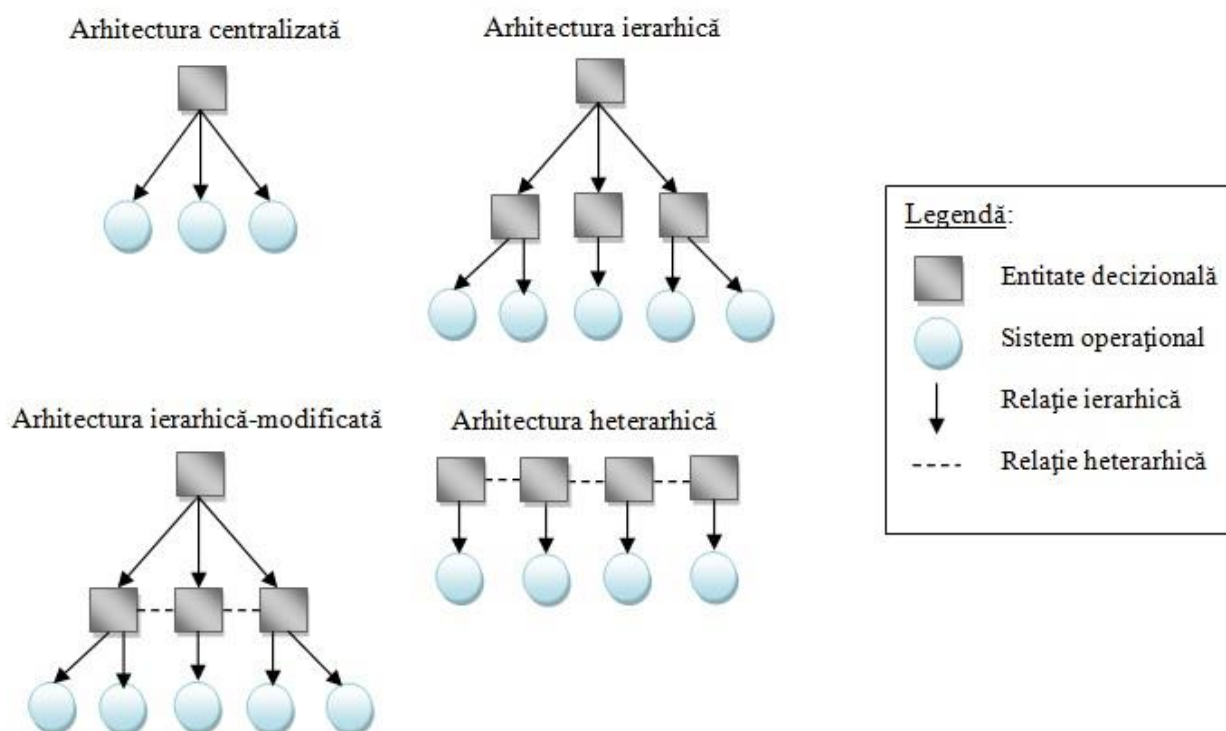


Fig. 2.1. Cele 4 tipuri de arhitecturi de conducere a proceselor de fabricație (shop-floor)

Existența unui singur element decizional, precum și ușurința accesului către informații globale oferă beneficii ca posibilitatea optimizării producției la nivel global și existența unei singure surse de informații.

Sistemele tradiționale sunt rigide, deoarece trecerea de la fabricarea unui produs la altul necesită timp, un volum de muncă și cheltuieli mari.

Dezavantajele sistemelor centralizate sunt datorate: posibilitatii blocarii sistemului la aparitia unei defectiuni (*single point of failure* – SPOF), vitezei cu care sunt procesate informațiile - în special în situații în care sistemul conține un număr mare de unități operaționale, și complexitatii software a procesorului central. În cazul în care în sistem apar perturbatii, informațiile legate de acestea trebuie să fie transmise de la nivelul operațional la unitatea centrală care le procesează și transmite mai apoi comenzi înapoi către nivelul operațional. Timpul în care sistemul detectează și reacționează la perturbatii poate fi în unele cazuri foarte mare, de aceea alegerea arhitecturii de tip centralizată nu este recomandată în cazul în care frecvența apariției perturbatiilor în sistemul de producție este mare.

Deși sistemele centralizate oferă o soluție aproape optimă în medii statice și deterministe, din punct de vedere al planificării producției, aceste sisteme nu pot reacționa suficient de rapid la perturbatii și nu pot evolua odată cu mediul în care funcționează. Din acest motiv, s-a făcut trecerea către alte tipuri de arhitecturi.

O astfel de arhitectură, care își propune să elimine deficiențele arhitecturii prezentate mai sus este cea ierarhică, în care sunt definite mai multe niveluri de control, pe o structura piramidală. În vârful ierarhiei se află elementul decizional, specific arhitecturilor centralizate, care se ocupă cu luarea deciziilor la nivel global și stabilirea obiectivelor globale ale sistemului.

Aceste decizii sunt transmise apoi către nivelurile ierarhice inferioare, acestea având rolul de a controla nivelul operațional. Pe măsură ce comenzile sunt transmise către nivelurile inferioare, nivelul de detaliu al informațiilor crește.

Avantajele utilizării unei astfel de arhitecturi sunt scăderea complexității software, timp de răspuns mai rapid și posibilitatea adăugării elementelor de control intermediar. Însă, existența mai multor niveluri de control sporește complexitatea comunicației dintre diferitele niveluri.

În acest tip de conducere, relația dintre diferitele niveluri de control este de tip master – slave. Această relație este relaxată în arhitectură de tip ierarhic-modificat. Principala diferență între arhitectură de tip ierarhic și cea ierarhică-modificată este că în cea din urmă nivelurile de control subordonate nivelului central cooperează între ele pentru a executa comenzile primite de la nivelul superior. Acest lucru denotă unul dintre beneficiile acestui tip de arhitectură, și anume că nivelurile intermediare au un anumit grad de autonomie. Dezavantajele sunt creșterea dificultății proiectării sistemului și a conectivității dintre elemente.

Discrepanța dintre producția planificată off-line și cea actuală, în timp real, precum și lipsa de reacție sau reacția prea înceată a sistemelor ierarhice la perturbatii precum comenzi prioritare, defectarea unor resurse din sistem, epuizarea stocului de piese sau introducerea unor noi produse sunt motivele principale pentru care interesul față de sistemele distribuite a crescut în ultima perioada.

În (McFarlane et al.,2002) sunt prezentate caracteristicile pe care sistemele industriale distribuite ar trebui să le aibă: (i) există mai multe elemente implicate în funcționarea unui sistem, (ii) unul sau mai multe elemente sunt separate din punct de vedere fizic sau logic, (iii) funcționalitățile/capabilitățile sistemului sunt îndeplinite de către mai multe elemente și (iv) sarcinile sistemului sunt executate de mai multe elemente. Specific arhitecturii distribuite este caracterul auto-organizatoric și auto-adaptiv al elementelor din sistem.

Creșterea interesului față de sistemele distribuite, precum și dezvoltarea domeniului comunicațiilor a determinat propunerea arhitecturii heterarhice. Un sistem heterarhic este format din entități autonome distribuite care comunica și cooperează între ele pentru a-și îndeplini obiectivele sau pentru a îndeplini obiectivul global al sistemului de fabricație. Între aceste entități există o relație de la egal la egal (peer-to-peer), spre deosebire de arhitectura ierarhică în care între nivelurile de control există o relație de tip master-slave.

Lipsa elementului central și a informațiilor globale este compensată prin capacitatea entităților din sistem de a coopera și de a lua decizii pentru a îndeplini o sarcină specifică și prin capacitatea de a stoca și prelucra informații la nivel local. Fiecare entitate decizională din sistem are acces doar la o parte din informații și de aceea pentru a-și îndeplini obiectivele comunica și cooperează cu celelalte entități din sistem.

Unul dintre avantajele sistemelor heterarhice provine din transformarea sistemului centralizat într-un sistem distribuit care conține mai multe sisteme de calcul, ceea ce implică reducerea complexității sistemului software, reconfigurabilitate și toleranță la defect.

Un dezavantaj important pentru arhitectură de tip heterarhic este lipsa perspectivei globale a entităților distribuite, fenomen numit miopie. Soluția pentru evitarea acestui fenomen în unele propuneri de sisteme heterarhice, spre exemplu PROSA și ADACOR, este introducerea unei entități numite supervisor care are perspectiva globală asupra sistemului. Supervisorul propune soluții pentru planificarea producției, iar entitățile din sistem decid dacă le vor urma sau nu.

Un alt dezavantaj al sistemelor heterarhice este dat de faptul că sistemul de comunicație trebuie să fie scalabil și să permită entităților să comunice în timp real.

Sistemele de conducere centralizate sunt utilizate cu precădere în cazurile în care optimizarea producției este o cerință importantă, iar frecvența apariției perturbațiilor este scăzută sau timpul de răspuns al sistemului nu este important.

Analizând evoluția tipurilor de arhitecturi, se poate observa că tendința în sistemele de conducere este de a minimiza utilizarea informațiilor globale și a elementelor centrale, de a coborî nivelul decizional cât mai aproape de nivelul operațional și de a oferi cât mai multă autonomie entităților decizionale prin relaxarea relațiilor de tip master-slave dintre nivelurile de control.

Dezvoltarea tehnologiei a produs schimbări în cerințele din industrie, făcând trecerea de la criterii de performanță de optimizare la necesitatea de adaptabilitate, vizibilitate și reactivitate la schimbări. În momentul de față se pune accent mai curând pe soluții satisfăcătoare, adaptabile și robuste decât pe soluții optime.

2.3. Obiective propuse

Obiectivul acestui studiu este de a propune o soluție de conducere de tip heterarhic centrată pe fluxul de produse din sistemul de fabricație care să asigure flexibilitate și robustețe, precum și testarea și validarea acesteia pe o platformă industrială de tip job-shop.

Arhitectura de conducere propusă este de tip holonic și conține ca referință arhitectura PROSA (*Product-Resource-Order-Staff Architecture*, Van Brussel et al, 1998). Un sistem de tip holonic este compus din entități autonome, capabile să comunice și să coopereze, numite holoni. Un holon este alcătuit dintr-o parte informațională, capabilă să proceseze date și, de cele mai multe ori, dintr-o parte fizică.

Arhitectura de conducere aleasă este formată din următorii holoni de baza: Holon Resursă (HR), Holon Produs (HP) și Holon Ordin (HO). Fiecare holon din sistem este responsabil pentru anumite aspecte din procesele de fabricație, precum gestionarea informațiilor legate de procesul de producție, corelarea acestor informații cu capacitățile resurselor din sistem și execuția operațiilor de procesare.

Soluția de conducere este un sistem de tip heterarhic care conține 2 moduri de execuție. În primul caz planificarea operațiilor și alocarea resurselor se fac înainte de începerea execuției, adică înainte de intrarea produsului în celula de fabricație, iar în cel de-al doilea caz alocarea resurselor se face înaintea următoarei operații.

Soluția propusă corespunde cerințelor actuale ale sistemelor de fabricație flexibile. Astfel, arhitectura sistemului de conducere propusă asigură flexibilitate, robustețe, toleranță la defect și autonomie. Flexibilitatea este datorată faptului că introducerea unor noi entități (resurse, produse, comenzi etc.) în sistem nu determină oprirea totală a producției, iar reconfigurarea elementelor din sistem se poate face cu ușurință. Robustețea sistemului este dată de reacția sistemului la apariția perturbațiilor. În momentul detectării unei perturbații (defectarea unei resurse sau epuizarea stocului de piese al unei resurse) produsele aflate în acel moment în celulă de fabricație își adaptează automat planul de alocare a resurselor pentru operații.

Automatizarea dirijată de produs este asigurată prin implementarea conceptului de **produs inteligent**. În momentul în care în sistem este plasată o comandă, fiecărui produs

corespunzător comenzii îi este atribuit un agent software capabil să comunice cu alte entități din sistem, să decidă asupra propriului parcurs și să aibă capacități de stocare. Astfel sunt îndeplinite caracteristicile ce definesc un produs inteligent.

Implementarea conceptului de produs inteligent în această lucrare constă în atribuirea unui Dispozitiv Inteligent Îmbarcat - IED (*Intelligent Embedded Device*) fiecărui produs care urmează să fie fabricat. IED-ul are capacitatea de a stoca informația și de a suporta execuția unui agent software și a comunicației de tip wireless pentru conectivitatea cu celelalte entități de tip agent.

Considerând structura generică de tip job-shop a celulei de fabricație studiată, în care este testat sistemul propus și posibilitatea execuției unei operații de către mai multe resurse, produsul inteligent este responsabil pentru planificarea operațiilor și alocarea resurselor din sistem. Acest lucru este posibil în urmă unui dialog între resurse și produsele inteligente, iar criteriul folosit în procesul de alocare a resurselor pentru operații este gradul de ocupare al resurselor.

Sistemul holonic prezentat în această lucrare este implementat prin intermediul sistemelor multi-agent folosind platforma JADE (*Java Agent Development Environment*). Această platformă este propusă din următoarele motive: este o platformă open-source, conformă cu specificațiile FIPA și prezintă funcționalitățile necesare implementării unui sistem distribuit bazat pe produs inteligent, cum ar fi comportamente – agenții au capacități decizionale și asigurarea comunicației dintre agenți prin intermediul mesajelor.

3. Conceptul de produs inteligent

Produsul este nucleul de baza al lanțurilor de aprovizionare și al sistemelor de fabricație. Pornind de la această idee și datorită tendinței actuale de orientare a pieței către client, conceptul de produs inteligent este tot mai frecvent regăsit în literatură de specialitate.

Noțiunea de produs inteligent poate fi aplicată în mai multe domenii, precum: producție, managementul lanțurilor de aprovizionare și logistică și managementul ciclului de viață al produsului.

Conceptul de produs inteligent a fost descris de mai mulți autori, dintre care amintim:

- **Wong et al., 2002**

Autorii consideră că un produs poate fi considerat inteligent dacă are o parte dintre sau toate cele 5 caracteristici enunțate mai jos:

1. Posedă un identificator unic
2. Este capabil să comunice într-un mod eficient cu mediul în care se află
3. Poate stoca informații despre el
4. Implementează un limbaj pentru a afișa caracteristicile sale, cerințele de producție etc
5. Este capabil să participe în procesul de luare a deciziilor sau chiar să ia decizii privitoare la destinul sau.

Pornind de la ideea că un produs inteligent reprezintă legătură dintre un produs fizic și reprezentarea informațională a să, autorii explică conceptul de produs inteligent prin exemplul unui borcan de sos (Fig. 3.1.).

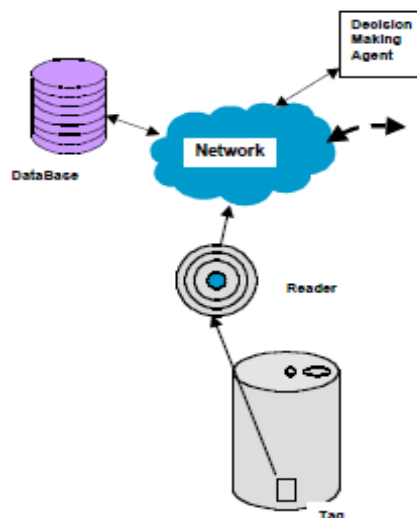


Fig.3.1. Borcan inteligent de sos

În acest exemplu, borcanul reprezintă produsul fizic din sistemul de fabricație sau din lanțul de aprovizionare. Produsul fizic conține o etichetă sau un identificator unic care poate fi scanat de un cititor de coduri. Acest identificator unic face legătura dintre produsul fizic și informațiile stocate în baza de date. Inteligența acestui produs este dată de asocierea unui agent software care este capabil de a lua decizii.

Adițional caracteristicilor definite mai sus, autorii au definit două niveluri de inteligență:

Nivelul 1 – un produs care este capabil să comunice starea sa, cum ar fi: formă, compoziție, localizare, caracteristici de baza etc. Acest produs este orientat către informație și are caracteristicile 1-3,

Nivelul 2 – un produs care este capabil să-și evalueze și să-și influențeze modul de funcționare. Acest tip de produs are caracteristicile 1-5 și este orientat către decizie.

Se consideră că produsele din prima categorie vor aduce avantaje în sistem pe termen scurt (2-5 ani), iar produsele din cea de-a doua categorie vor aduce beneficii pe termen lung (5-10 ani).

- **Kärkkäinen et al., 2003**

Autorii consideră că un produs inteligent influențează starea produsului în interiorul lanțului de aprovizionare și ciclul sau de viață. Acest lucru înseamnă că un produs inteligent controlează traseul pe care îl parcurge și modul în care trebuie să fie manevrat. Caracteristicile unui produs inteligent sunt:

1. Are un identificator unic global
2. Conține o legătură către informațiile sale; acestea pot fi conținute în codul de identificare sau pot fi accesate prin mecanisme de căutare
3. Este capabil să informeze sistemul și utilizatorii cu privire la starea sa (chiar și proactiv).

Diferența dintre definiția dată de Wong et al. și cea dată de Kärkkäinen et al. este că cea de-a doua definiție se referă cu precădere la inteligență integrată în produs, în schimb ce prima definiție se referă la inteligență adăugată produsului. Un punct comun în cele două definiții este utilizarea tehnologiei RFID pentru înzestrarea produsului cu identitate.

- **Ventă, 2007**

Ventă consideră că un produs inteligent trebuie să aibă fie capabil să:

1. Monitorizeze în permanență starea și mediul în care se află
2. Reacționeze și se adapteze la mediul și condițiile de funcționare
3. Mențină o funcționare optimă chiar și în situații de excepție
4. Comunice într-un mod eficient cu utilizatorii, mediul, alte produse și cu sistemul.

Această definiție este centrată pe capacitatea produsului de a lua decizii și pe inteligența integrată. Autorul presupune că produsul are suficientă capacitate de procesare integrată pentru a putea comunica direct cu celelalte entități din sistem.

Diferența dintre cele 3 definiții ale produsului inteligent constă în modul în care este gândită inteligența sau mai degrabă locul în care se afla inteligența. Prima definiție caracterizează un produs a cărui inteligență este în rețea, iar ultimele două definesc o inteligență integrată în produs care îi dă o flexibilitate mai mare.

Toate cele trei definiții subliniază necesitatea produsului de a avea o identitate, capacitate de a comunica direct sau indirect cu celelalte entități din sistem (alte produse, resurse, utilizatori etc) și de a lua decizii cu privire la parcursul lor prin lanțul de aprovizionare.

În (Meyer et al., 2009) au fost definite trei criterii de clasificare a produselor inteligente. Criteriile sunt:

- *După nivelul de inteligență:*
 - Manipularea informației: un produs trebuie să fie capabil să manipuleze propriile sale informații provenite de la senzori, cititoare de coduri RFID sau alte tehnologii. Fără aceasta capacitate un produs nu poate fi considerat inteligent.
 - Notificarea problemelor: un produs poate să notifice utilizatorul în momentul apariției unei probleme. Produsul nu poate controla propriul destin, dar poate raporta când intervin schimbările în starea lui.
 - Luarea deciziilor: un produs poate controla propriul parcurs și poate decide asupra destinului său fără intervenții din exterior.
- *După localizarea inteligenței:*
 - Inteligența prin rețea: inteligența este în afara produsului fizic; produsul conține un dispozitiv care e folosit ca interfață pentru inteligență (se mai numesc platforme portal).
 - Inteligența incorporată în obiect: toată inteligența se află pe produsul fizic; pentru

aceasta produsul trebuie să aibă capacitate de procesare, de stocare a informației și trebuie să se poate conecta la rețea (se mai numesc platforme integrate).

- *După nivelul agregat de inteligență* (pentru produsele compuse din mai multe elemente):
 - Element inteligent: poate manipula informații, notifica și lua decizii pentru el; dacă conține alte componente, acestea nu pot fi considerate obiecte individuale
 - Container inteligent: poate face aceleași lucruri ca un element inteligent însă părțile componente pot acționa ca obiecte inteligente individuale. Dacă containerul inteligent este dezasamblat sau unele parti sunt inlaturate, fiecare din componentele sale continua sa fie un produs/container inteligent.

În (McFarlane et al., 2013) este evidențiat faptul că definițiile de mai sus descriu mai curând caracteristicile pe care un produs inteligent trebuie să le aibă decât ceea ce înseamnă. Definiția propusă de autorii de mai sus este:

Un produs inteligent este o comandă fizică sau produs care este legat de informațiile și regulile privind procesul de producție, stocare sau transport al produsului și care oferă produsului abilitatea de a influența aceste operații.

Dezvoltările recente în domeniul auto-identificării, a sistemelor distribuite și a dispozitivelor integrate (embedded devices) au avut un impact mare în creșterea interesului față de conceptul de produs inteligent, în special pentru implementarea acestuia în industrie și pentru evidențierea beneficiilor pe care produsul inteligent le poate aduce.

La începutul acestui studiu a fost evidențiat faptul că produsul inteligent are aplicabilitate în mai multe domenii, precum: sisteme de fabricație și lanțuri de aprovizionare. În continuare vor fi prezentate beneficiile majore pentru fiecare dintre aceste domenii.

În domeniul sistemelor de fabricație, implementarea conceptului de produs inteligent poate aduce beneficii în planificarea producției, personalizarea produselor și introducerea de noi produse.

În (Meyer et al., 2010), este evidențiat faptul că utilizarea conceptului de produs inteligent poate duce la creșterea robusteții în sistemele de control, deoarece perturbatiile pot fi tratate la nivel local, spre deosebire de sistemele centralizate și deasemenea poate aduce beneficii asupra managementului perturbatiilor. Autorii evedentiaza motivele pentru care tratarea perturbatiilor la nivel local este benefică.

Primul motiv este că timpul în care un sistem centralizat tratează problemele locale este întotdeauna mai mare decât în cazul unui sistem distribuit.

Al doilea motiv este acela că pentru un sistem centralizat este dificil de luat în considerare toate constrângerile entităților individuale. Această afirmație evidențiază faptul că sistemele distribuite și implicit, produsele inteligente sunt o soluție viabilă pentru personalizarea producției, adică orientarea producției către client. Al treilea motiv prezentat de autori este existența unui singur punct de defectare (SPOF) în sistemele centralizate.

Personalizarea producției, adică fabricarea unor produse care variază de la o comanda la alta presupune că produsele să aibă diferite rute prin celula de fabricație și diferite operații și materiale prime să fie folosite în procesul de fabricație. Atribuirea unui produs inteligent pentru

fiecare produs fizic ce urmează să fie fabricat facilitează optimizarea rutei pe care o parcurge produsul prin celula de fabricație și deasemenea dirijarea materiilor prime.

În contextul lanțurilor de aprovizionare, produsele inteligente pot aduce beneficii în livrarea și securitatea produselor. Astfel, trasabilitatea produselor îmbunătățește timpul de livrare a bunurilor și, în mod evident, facilitează monitorizarea stocurilor. Prin atribuirea inteligenței unui produs fizic, ruta de livrare a produsului către destinatar poate fi modificată în timp ce produsul este în tranzit, iar securitatea produselor este îmbunătățită deoarece autenticitatea pachetele poate fi lesne verificată și poate fi descoperit locul în care s-a produs o anomalie.

În ultima perioada, rețeaua *stakeholderilor* din lanțurile de aprovizionare a devenit din ce în ce mai complexă, iar numărul companiilor implicate în producția unor bunuri este în creștere. Pentru succesul unui astfel de sistem trebuie să existe comunicare și cooperare între organizațiile partenere, iar fiecare stakeholder trebuie să aibă acces către informațiile legate de bunurile lansate pe piață.

Prin utilizarea conceptului de produs inteligent, schimbul de informații între companii, precum și monitorizarea produselor din lanțul de aprovizionare se fac în timp real.

Ciclul de viață al produsului pasiv, în manieră tradițională, prezintă multe dezavantaje. Etapele din ciclul de viață al unui produs sunt: faza de început a vieții (BOL – *Beginning of Life*) care include etapele de proiectare și specificare a cerințelor, faza de mijloc vieții (MOL – *Middle of Life*) care include etapele de producție, livrare și utilizare a produsului de către consumator și faza de sfârșit a vieții (EOL – *End of Life*) care include etapele de mentenanță și dezasamblare sau reciclare a produsului. Aceste etape trebuie să fie strâns legate una de cealaltă pentru a putea duce la îmbunătățirea produselor. Spre exemplu, costurile de mentenanță ale unui produs ar putea fi minimizate dacă s-ar cunoaște cauza deteriorării lui, cum ar fi utilizarea neadecvată sau o calitate proastă. Realizând asocierea dintre produsul fizic și informațiile obținute de-a lungul ciclului de viață al produsului, pot fi îmbunătățite etapele de proiectare, producție și mentenanță, reducând costurile de producție și, posibil, crescând gradul de satisfacție al clienților.

În (McFarlane et al., 2013), autorii evidențiază faptul că pentru a demonstra beneficiile pe care le aduce implementarea produselor trebuie definite scenarii de utilizare a acestuia. În articol sunt definite 2 categorii de scenarii: statice și dinamice. Scenariile statice sunt:

- i) în situația în care un produs sau o comandă trece prin mai multe organizații în procesul de livrare;
- ii) când un element specific face parte din mai multe comenzi pe parcursul producției sau a livrării;
- iii) dacă cerințele specifice ale unui client diferă de intențiile companiei de logistică;
- iv) în cazul în care o comandă este segmentată, iar fiecare parte din acea comandă este controlată de diferite organizații;
- v) când stakeholderii din lanțul de aprovizionare au acces parțial sau deloc la informații.

Dintre scenariile dinamice amintim: (i) în situația în care rutarea produselor se face dinamic și există oricând posibilitatea apariției unei rute alternative; (ii) când apariția perturbatiilor este frecvența și performanțele sunt greu de obținut; (iii) în cazul în care în gestiunea comenzilor este necesară intervenție umană care nu este valabilă; (iv) când preferințele clienților se schimbă între efectuarea comenzii și livrarea ei; (v) dacă caracteristicile unui produs se schimbă în timp.

4. Soluția de conducere heterarhică propusă

4.1. Descriere generală a soluției

Soluția propusă în cadrul acestui studiu este un sistem de conducere de tip heterarhic (Fig. 4.1.) centrat pe fluxul de produse din sistemul de fabricație. În cadrul arhitecturilor de conducere de tip heterarhic, nivelul decizional se află cât mai aproape de nivelul operațional. În cadrul sistemului de conducere propus, elementul decizional este atașat nivelului fizic.

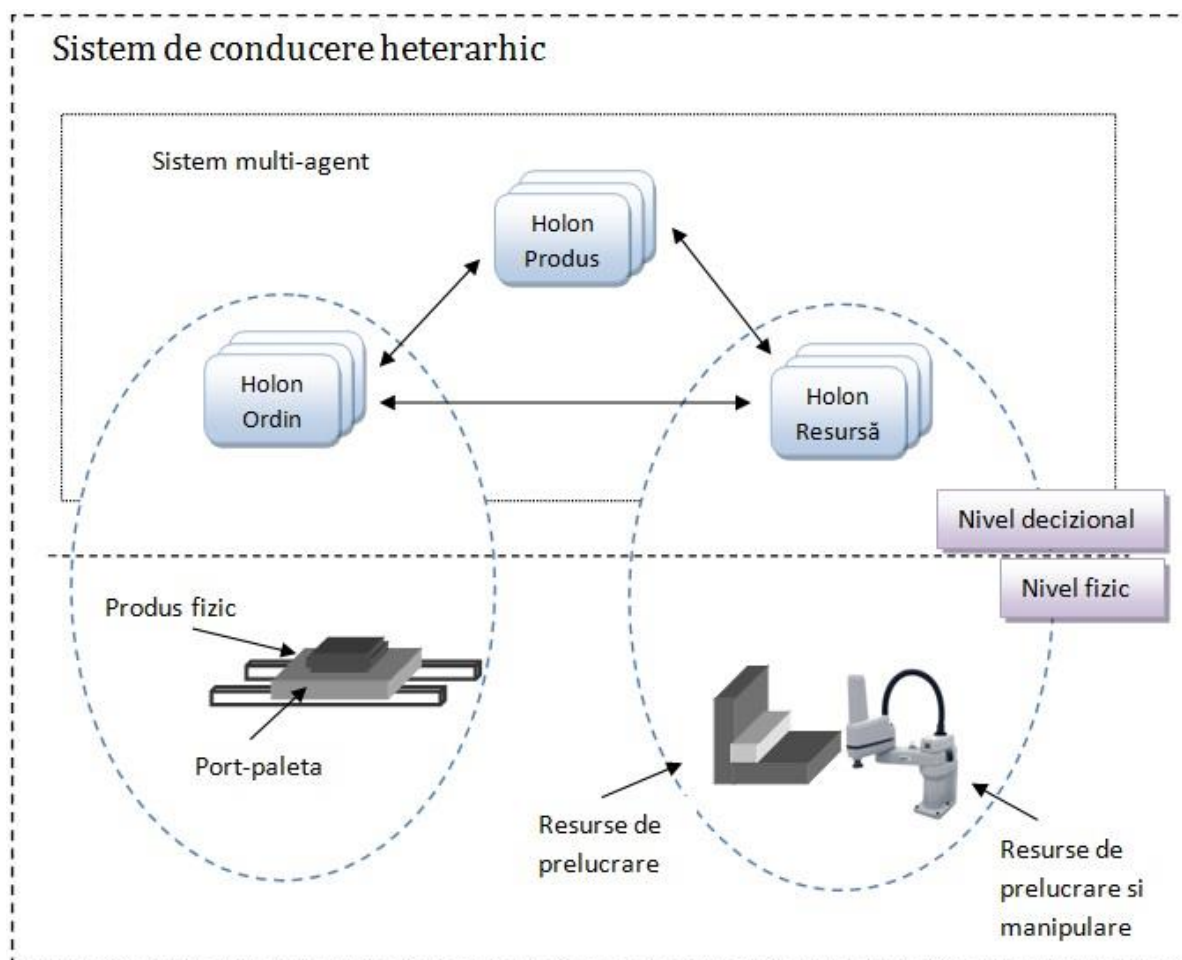


Fig. 4.1. – Sistem de conducere heterarhic

Sistemul de conducere heterarhic este un sistem holonic care are la baza elementele standard ale arhitecturii de referință PROSA (Product-Resource-Order-Staff Architecture). Holonii de bază sunt: Holon Produs, Holon Resursă și Holon Ordin. Holonul de tip Produs conține informațiile necesare procesului de fabricație, holonul de tip Resursă reprezintă fiecare resursă din celulă de fabricație și este formată din două componente: partea informațională, capabilă să prelucreze informații despre resursă și să controleze resursa fizică și partea fizică reprezentată de robotul sau mașina de prelucrare propriu-zisă. Holonul de tip Ordin conține starea în timp real a produselor ce sunt fabricate și este format din două componente: componentă informațională, responsabilă de alocarea task-urilor și componentă fizică, reprezentată de produsul inteligent.

Sistemul de conducere prezentat în această lucrare a fost testat și validat pe sistemul de fabricație din cadrul laboratorului de Robotică și Inteligență Artificială din cadrul Facultății de Automatică și Calculatoare. Infrastructura celulei de fabricație este de tip job-shop și este compusă din:

- 5 posturi de lucru
- Robot cartezian pentru alimentarea și evacuarea paletelor de transport
- 2 mașini de prelucrare (CNC)
- Sistem de transport
- Automat programabil

În Fig.4.2. este reprezentată structura unei celule de fabricație generice (proces shop-floor)..

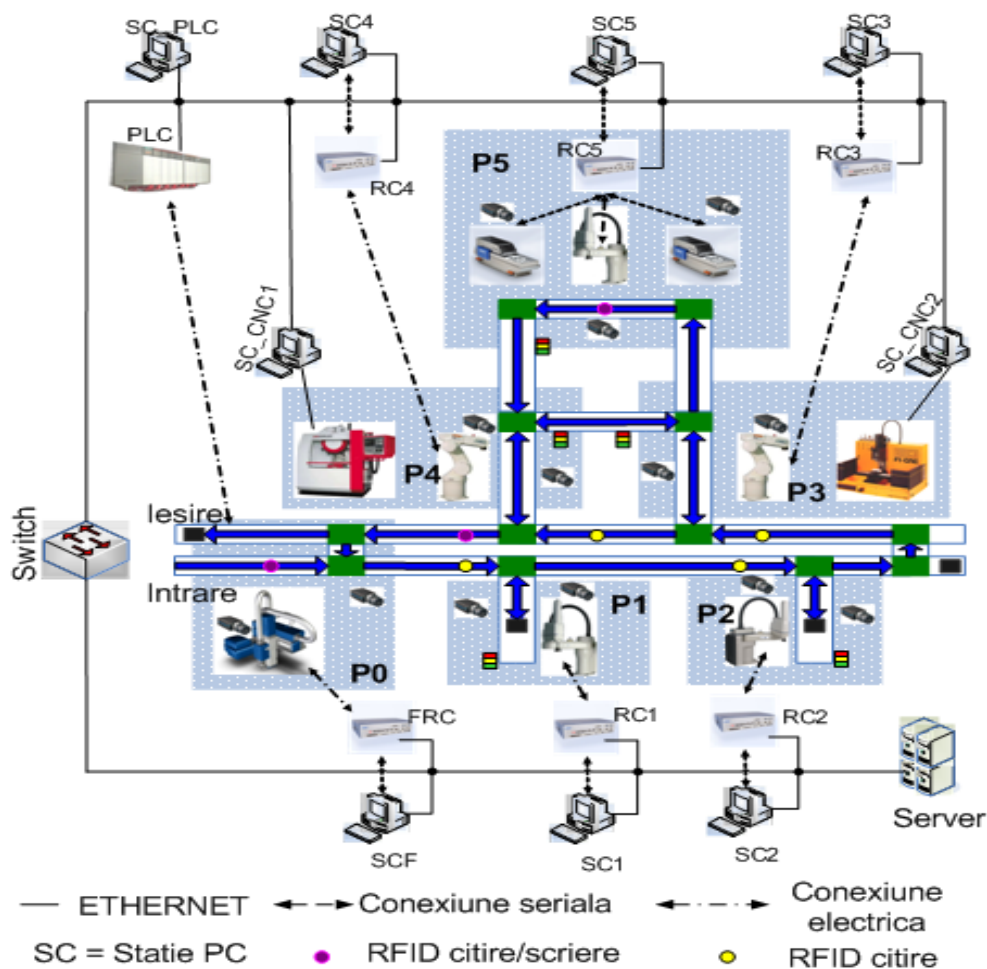


Fig. 4.2. – Infrastructura celulei de fabricație folosite pentru testarea soluției

Cele 6 posturi de lucru conțin un robot cartezian (P0), 2 roboți de tip articulat vertical (P3,P4) și 3 roboți de tip SCARĂ (Selective Compliance Articulated Robot Arm) (P1,P2,P5).

Resursa P0 este folosită pentru a introduce și a evacua port-paletele din sistem. Resursele P1-P5 sunt folosite pentru operații de manipulare și interacțiuni cu materialele, precum montaj, împachetare, înșurubare, alimentare/descărcare utilaje etc. Între resursele P1 și P2 se află un spațiu comun pentru manipularea obiectelor și astfel pot fi executate programe de cooperare, sincronizare și evitare a coliziunilor între cele 2 resurse. Resursa P5 conține 2 dispozitive de tip

feeder și poate executa operații de identificare a pieselor și poziționarea a acestora pe paletele de transport.

În celula de fabricație, sistemul de vedere artificială este compus din două tipuri de camere: staționare – camerele sunt montate pe tavanul celulei și mobile – camerele sunt montate pe brațul robot.

Sistemul de transport este folosit pentru transportul și transferul produselor prin celula de fabricație și între posturile de lucru. Acesta este compus dintr-un conveior longitudinal care transporta produsele într-o buclă închisă și un conveior transversal care face legătură dintre bandă principală și posturile de lucru. Sistemul de transport este dotat cu un model de evitare a coliziunilor. Dispozitivele port-paleta au rolul de a transporta paletele pe care sunt poziționate produsele.

Sistemul de identificare a produselor este bazat pe tehnologia RFID. Dispozitivele de tip port-paleta sunt echipate cu etichete de tip RFID care pot fi citite și scrise, iar de-a lungul conveiorului longitudinal sunt dispuse capete de citire/scriere (fig. 4.2.) pentru a putea fi identificate produsele.

Tipurile de operații posibile în celulă de fabricație descrisă mai sus sunt: asamblare și montaj, prelucrări pe mașini de tip CNC, control de calitate utilizând sisteme de vedere artificială.

Componentele de control din celulă de fabricație sunt: stații PC asociate fiecărei resurse din celulă prin care sunt comandate mașinile de prelucrare și roboții industriali, automat programabil de tip Bosch-Rexroth care comandă sistemul de transport, execuția operațiilor și trasabilitatea produselor și controllerelor robot care comandă resursele de tip robot industrial și integrează sistemul de vedere artificială.

Toleranța la defect a sistemului de conducere din celula de fabricație este realizată prin sistemul de comunicație reprezentat în Fig. 4.3. Există 4 medii de comunicare:

- *Ethernet* – legătură dintre cele 3 elemente de conducere este realizată prin intermediul rețelei de tip Switched Ethernet. Dacă una dintre stațiile PC se defectează, o altă stație din cele funcționale poate prelua taskurile stației defectate, datorită faptului că datele sunt replicate între stații.
- *Comunicație serială* – între stațiile PC și controllerele robot există legăturile seriale directe prin intermediul cărora sunt monitorizate constant controllerele
- *Ring Inter Robot Controller*: în cazul în care Swith-ul nu este funcțional, conexiunea dintre stațiile PC și controller și între controller este asigurată prin acest timp de comunicație
- *Point-to-Point I/O* – realizează conexiunea între controllerele robot și este folosit pentru operații de sincronizare între roboți.

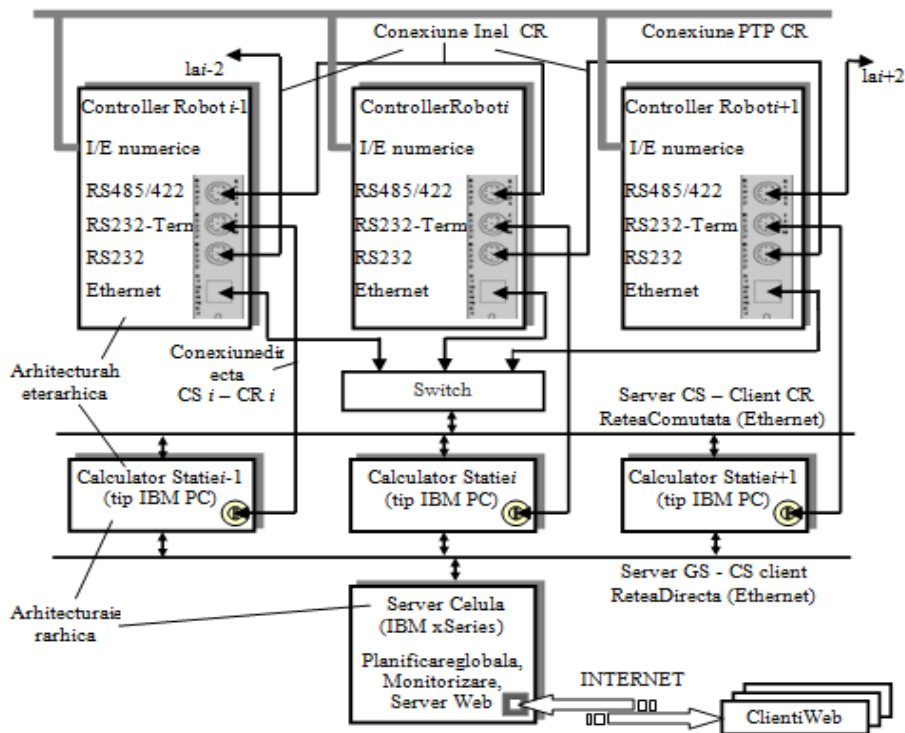


Fig. 4.3. – Sistemul de comunicație din celula de fabricație

Arhitectura heterarhica a sistemului de conducere și implementarea conceptelor de produs inteligent și automatizare dirijată de produs asigură funcționalități ca: robustețe, flexibilitate, toleranță la defect și reconfigurabilitate.

În sistemele care implementează automatizarea dirijată de produs, produsele sunt entități active în procesul de fabricație, fiind implicate în luarea deciziilor referitoare la procesul de fabricație. Astfel, răspunsul sistemului de conducere în momentul apariției perturbațiilor este mult mai rapid decât în cazul unui sistem cu o arhitectură centralizată.

Un alt beneficiu este faptul că desincronizarea dintre fluxul de materiale (produse fizice) și fluxul informațional, specifică sistemelor de conducere tradiționale, este eliminată în cazul sistemelor cu arhitectură heterarhică și a automatizării dirijate de produs, pentru că în acest caz componenta informațională este permanent atașată de componentă fizică.

Sistemul de conducere asigură robustețe în cazul apariției perturbațiilor de tipul defectare resurse sau epuizare stoc de piese. În momentul detectării perturbației de către sistemul de conducere produsele aflate în fabricație își adaptează automat planul de alocare a resurselor pentru operații. Structura modulară a sistemului de tip holonic asigură reconfigurabilitate și flexibilitate, astfel încât în momentul în care se dorește adăugarea sau modificarea unor componente în sistemul de conducere, cum ar fi adăugarea sau reconfigurarea unor noi resurse, adăugarea unor noi produse sau comenzi, procesul de producție nu necesită a fi întrerupt.

În cadrul sistemului propus, produsul inteligent este responsabil pentru alocarea operațiilor pe resurse și monitorizarea execuției. În procesul de alocare, produsul inteligent este responsabil pentru selectarea resurselor pentru fabricație. În algoritmul de selecție pot fi folosite diferite criterii astfel încât să se asigure o producție optimă.

Soluția de conducere conține 2 moduri de execuție. În primul caz, planificarea operațiilor și alocarea resurselor se fac înainte de începerea execuției, adică înainte de intrarea produsului în celulă de fabricație. Astfel, înainte de intrarea în execuție, produsul, reprezentat de holonul ordin (OH) calculează setul complet de operații și alocarea lor pe resurse și este cunoscută de la început ruta lui prin celulă de fabricație. Procesul de planificare a operațiilor și de alocare a resurselor se face în urma unui dialog între ordine și resursele valide din sistem. Criteriul de selecție al resurselor este încărcarea minimă a resursei.

Acest mod aduce beneficii datorită faptului că timpul total de producție a produsului respectiv poate fi minimizat, în cazul în care timpul de procesare și durata transportului sunt folosite ca criteriu de selecție. Dezavantajul major al acestui mod de funcționare este că, în cazul defectării unei resurse, producția este întreruptă și toate ordinele afectate de această perturbație reiau procesul de alocare a resurselor.

În cel de-al doilea mod de funcționare, alocarea resurselor se face înaintea următoarei operații. Astfel, rutarea produsului prin celulă de fabricație nu este cunoscută încă de la început.

Performanțele sistemului de conducere în prezența perturbațiilor sunt mai bune în acest mod de funcționare datorită faptului că în procesul de alocare a resurselor sunt luate în considerare doar acele resurse care sunt funcționale. Dezavantajul acestui mod de execuție este că în procesul de alocare, produsul nu are o perspectivă globală asupra performanțelor producției (timp total de execuție a operațiilor) și, prin urmare, producția nu este optimizată.

Selecția modului corespunzător de execuție dintre cele două propuse se face în funcție de caracteristicile sistemelor de fabricație și a cerințelor de business. Primul mod de execuție poate fi folosit în situația în care frecvența apariției perturbațiilor este mică și se impun cerințe de performanță. Al doilea mod de execuție poate fi folosit în cazul în care sistemul de fabricație este deseori afectat de perturbații și performanța sistemului nu este o prioritate.

4.2. Structura sistemului multi-agent

Arhitectură heterarhică a sistemului de conducere este implementată printr-un sistem multi-agent care conține elementele standard ale arhitecturii de referință PROSA: holon resursă, holon ordin și holon produs.

Holonul de tip resursă (RH) este format dintr-o parte fizică și o parte informațională. Componentă fizică a holonului este reprezentată de resursele de producție din celulă de fabricație, adică roboți industriali, mașini de prelucrare etc. Componenta informațională a holonului are capacitate de stocare și prelucrare a informațiilor legate de starea resurselor din sistem și controlează resursă fizică.

Holonul de tip produs (PH) conține informații necesare fabricației fiecărui tip de produs, că lista materialelor, ciclul de viață al produsului, rețeta produsului, cerințele clientului și cerințele de calitate. Holonul produs este un holon pasiv care acționează ca un server de informații pentru ceilalți holoni din sistem.

Holonul de tip ordin (OH) este reprezentarea în timp real a produselor ce sunt fabricate. OH este responsabil pentru monitorizarea producției și execuția corectă a operațiilor. Holonul ordin este format dintr-o parte informațională și o parte fizică. Componenta fizică a holonului de tip ordin este produsul fizic care urmează să fie fabricat, iar componenta informațională este dată de modulul decizional din structura produsului inteligent și conține informații despre starea produsului și este responsabilă pentru planificarea și alocarea operațiilor pe resurse.

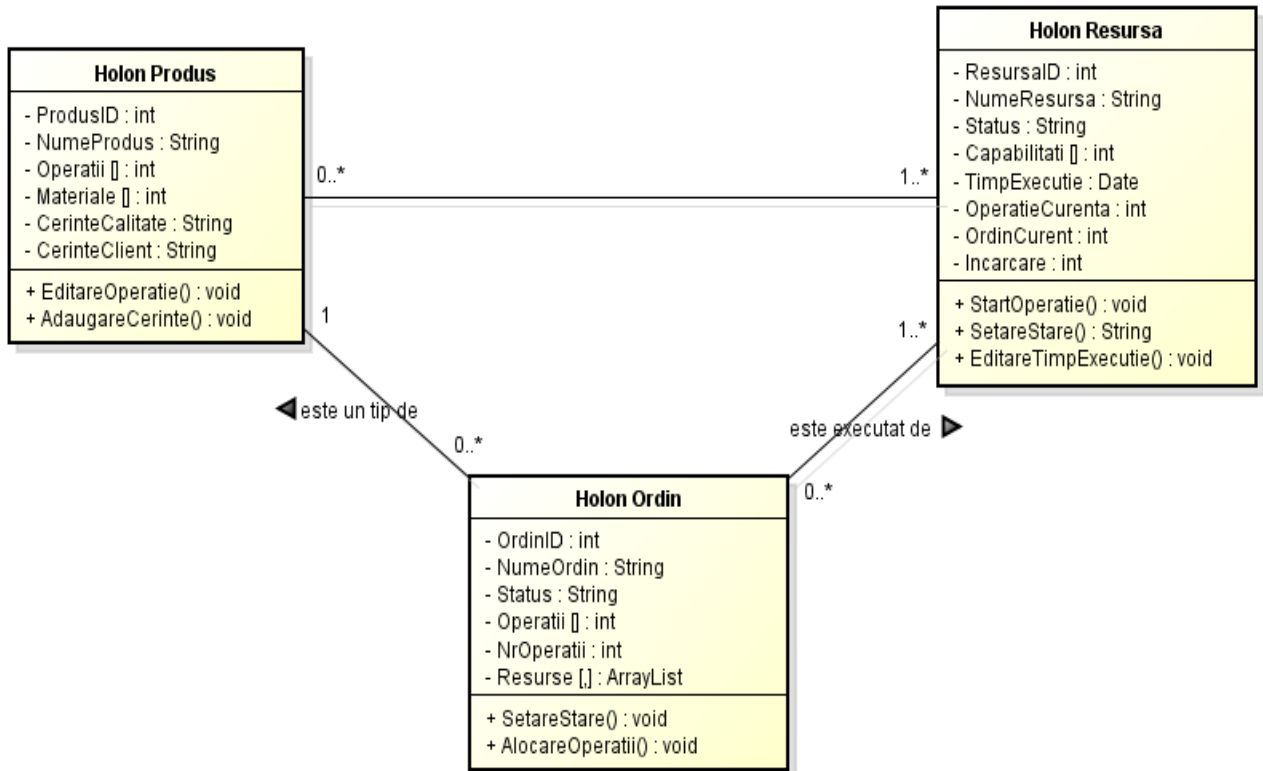


Fig. 4.4. – Diagrama de clasa – tipuri de date și funcții ale tipurilor de holoni

Fig. 4.4. reprezintă diagrama de clasa a sistemului multi-agent în care sunt definite tipurile de date și funcții ale fiecărui agent. Agentul de tip Produs conține lista de operații care sunt necesare fiecărui tip de produs, lista de materiale, cerințe de calitate și opțional, lista cerințelor clientului pentru produsele personalizate.

Agentul de tip Resursă conține statusul resursei, lista de operații pe care le poate executa (capabilități), timpul de execuție al fiecărei operații, încărcarea resursei și informații legate de procesul de execuție, cum ar fi ordinul și operația curentă. Funcționalitățile pe care le are agentul de tip Resursă este de a configura parametrii de producție.

Agentul de tip Ordin conține următoarele date: starea produsului fizic și a procesului de fabricație, lista operațiilor necesare și alocarea operațiilor pe resurse. Funcționalitățile îndeplinite de agentul de tip Ordin sunt modificarea informațiilor despre starea produsului și a procesului de fabricație și alocarea operațiilor pe resurse.

Interacțiunea dintre cele 3 tipuri de agenți este reprezentată în Fig.4.5. și poate fi descrisă astfel:

- Agent Produs – Agent Resursă: între cei 2 agenți are loc un schimb de informații legate de procesul de fabricație, cum ar fi modul în care o resursă trebuie să execute o anumită operație, cerințele de calitate pentru operații, parametrii de proces, etc.
- Agent Produs – Agent Ordin: între cei 2 agenți are loc un schimb de informații legate de fabricarea unui anumit produs, cum ar fi secvența operațiilor necesare, lista materialelor etc.

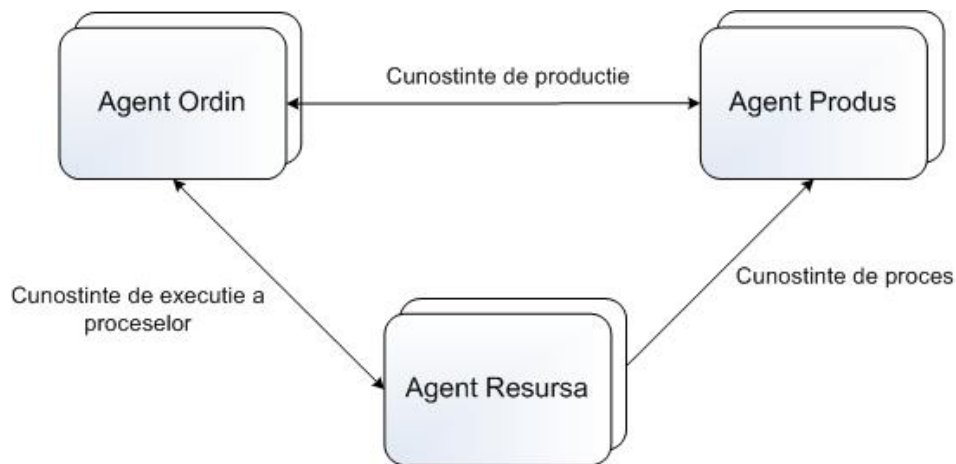


Fig. 4.5. – Interacțiune agenți

- Agent Ordin – Agent Resursă: între cei 2 agenți are loc un schimb de informații legate de stadiul execuției operațiilor pentru un anumit produs. Schimbul de informații are loc pentru procesul de alocare a resurselor pe operații, rezervarea resursei pentru operație, începerea și finalizarea execuției, monitorizarea execuției etc.

Ciclul de viață al agenților diferă în funcție de tipul de agent. Agentul de tip Produs este un agent pasiv folosit ca server de informații pentru celelalte tipuri de agenți, prin urmare, după activare așteaptă cereri de la agenții de tip Ordin pentru informații legate de execuție a producției. Ciclul de viață al agentului de tip Resursă este reprezentat în Fig.4.6. și începe cu etapa de activare. În etapa de așteptare, agentul Resursă interacționează cu agenții de tip Ordin, iar în urma interacțiunii comandă resursa fizică pentru a executa operații și/sau configurează parametrii de proces după care se întoarce în etapa de așteptare.

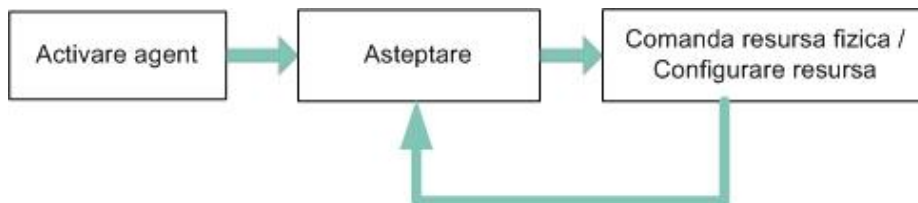


Fig. 4.6. – Ciclul de viață al agentului de tip Resursă

Ciclul de viață al agentului de tip Ordin este reprezentat în Fig.4.7. și începe cu etapa de activare, în care agentul primește informații despre procesul de fabricație a produsului, urmată de etapa de fabricație propriu-zisă în care agentul interacționează cu agenții de tip Resursă pentru alocarea resurselor pe operații și se finalizează cu etapa de suspendare. În cazul în care este necesară fabricația mai multor produse de același tip, agentul de tip Ordin este reactivat.



Fig. 4.7. – Ciclul de viață al agentului de tip Ordin

4.3. Comunicația și interconectarea agenților

Conexiunea componentelor de control în cadrul celulei de fabricație se face prin intermediul unei rețele de tip Ethernet. Mai exact, automatul programabil, stațiile PC și controllerele robot sunt conectate la un Switch Ethernet care oferă suport pentru comunicație wireless. Produsele inteligente din celula de fabricație folosesc comunicația de tip wireless care oferă mobilitate pentru a interacționa cu celelalte entități din sistem. Astfel, produsele se conectează prin intermediul unui punct de acces la Switch-ul Ethernet din sistemul de fabricație. Arhitectura de comunicație este reprezentată în Fig. 4.8. Interacțiunea componentelor de control se face prin intermediul adreselor IP.

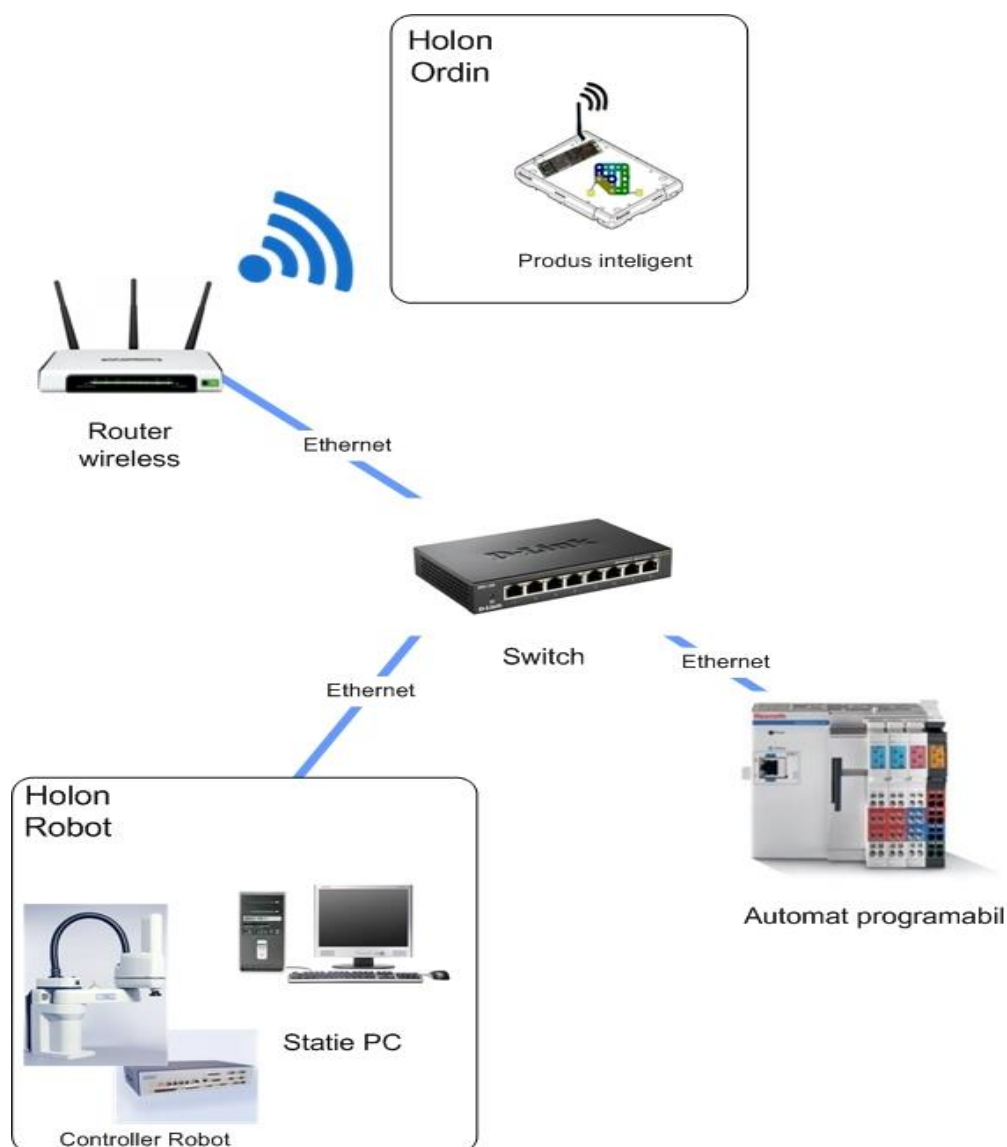


Fig. 4.8. – Schema de conexiune a componentelor sistemului multi-agent

Sistemul de comunicație folosit în implementarea sistemului multi-agent este reprezentat în schema generală de comunicație din Fig.4.9. Acesta conține următoarele componente:

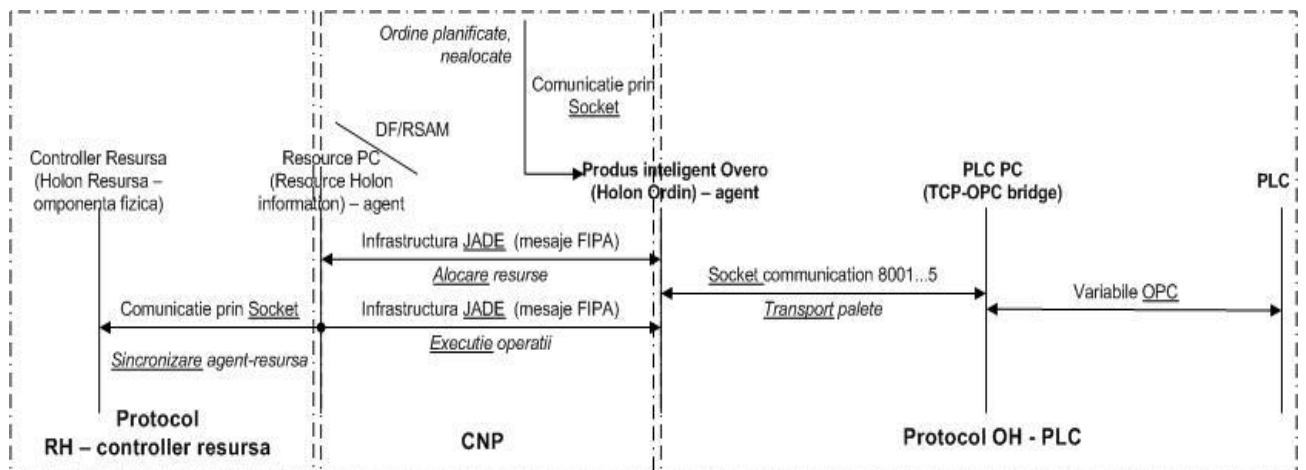


Fig. 4.9. –Schema generală de comunicație

- Comunicația dintre automat programabil și agenți de tip Ordin
- Comunicația între agenții de tip Ordin și agenții Resursă
- Comunicația dintre agenții de tip Resursă și Controllerele robot
- Comunicația dintre agenții de tip Resursă și controllerele robot

Fiecare resursă de tip robot industrial din celula de fabricație este reprezentată de un cotrespondent informational denumit agent resursă (RH) în sistemul multi-agent care este executat pe stația PC corespunzătoare resursei. Conexiunea dintre agentul resursă și robot se face prin intermediul controllerelor externe „Adept SmartController”.

Comunicația dintre agenții de tip Resursă și Controllerele robot (protocolul RH – Controller din Fig.4.9.) este realizată prin intermediul unei conexiuni de tip Ethernet, folosind protocolul TCP/IP.

- Comunicația dintre automat programabil și agenții de tip Ordin

Protocolul OH-PLC (Fig.4.10.) definește interacțiunea dintre automatul programabil și sistemul multi-agent, în special agenții de tip ordin (OH). Automatul programabil este folosit pentru transportul port-paletelor către posturile de lucru și pentru interogarea poziției port-paletei în celula de fabricație.

Citirea/scrierea datelor de intrare/ieșire din automatul programabil de către agentul de tip ordin se face prin intermediul aplicației TCP-OPC bridge. Interacțiunea dintre aplicația de tip agent și TCP-OPC bridge se face printr-un schimb de mesaje utilizând comunicația prin socket și protocolul de transport TCP/IP.

Un socket este un program software care stabilește o legătură bidirecțională între un program de tip server și unul sau mai multe programe de tip client. Adresa unui socket este dată de asocierea dintre o adresă IP și numărul portului. De regulă, socket-ul este creat în aplicația de tip server și este asociat unui port. Serverul, apoi așteaptă solicitări de la aplicațiile de tip client de conectare la socket-ul respectiv. În cadrul proiectului, clientul TCP/IP este încorporat în programul software al agentului, iar serverul TCP/IP este dezvoltat în aplicația TCP-OPC bridge.

Protocol OH - PLC

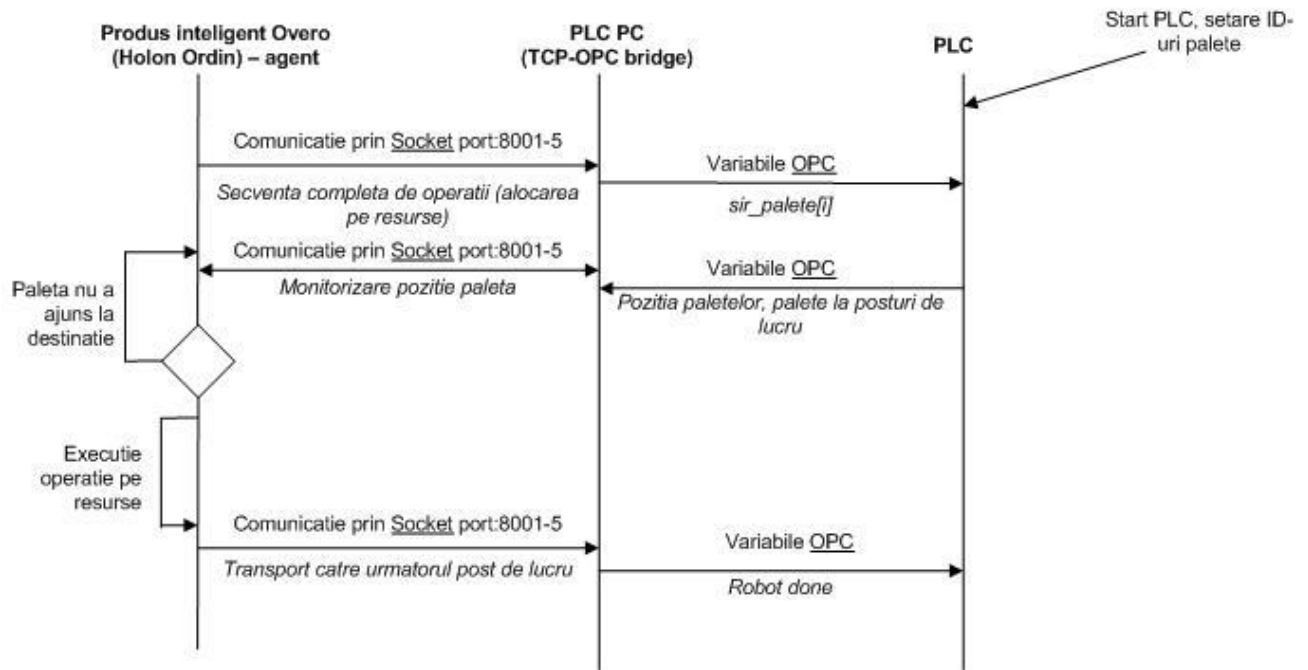


Fig. 4.10. – Detalierea protocolului OH –PLC

Interacțiunea dintre agentul Ordin și aplicația TCP-OPC bridge are loc în următoarele situații: (i) agentul Ordin trimite secvență completă a resurselor care trebuie vizitate, (ii) agentul Ordin monitorizează poziția port-paletei în celula de fabricație pentru a vedea dacă port-paleta se află în dreptul unui post de lucru și (iii) agentul Ordin notifica TCP-OPC bridge finalizarea operației la o anumită resursă pentru că port-paleta să fie transportată către următorul post de lucru. Descrierea pe larg a mesajelor și structura programelor software sunt prezentate în soluția de implementare.

Comunicația dintre TCP-OPC bridge și automatul programabil este realizată prin intermediul OPC. OPC (*Object Linking and Embedding for Process Control*) este un standard de comunicație în timp-real care permite interacțiunea dintre dispozitivele industriale hardware care provin de la producători diferiți.

OPC este implementat folosind arhitectura client/server. Serverul OPC este un program software care convertește protocolul de comunicație hardware a automatului programabil în protocolul OPC. Clientul OPC este un program software care necesită conectarea la automatul programabil. Datele din aplicațiile OPC sunt organizate în grupuri și item-uri.

- Interacțiunea dintre agenții de tip Ordin și agenții Resursă

Contract Net Protocol este un protocol folosit în sistemele multi-agent care se bazează pe distribuirea taskurilor (contractare și sub-contractare) între mai mulți agenți. Contract Net are la baza două tipuri de agenți: Inițiator și Participant. Într-un anumit moment, un agent poate fi inițiator, participant sau ambele. Inițiatorul sau managerul este agentul care inițiază licitația și care solicită un serviciu, iar participanții (contractorii) sunt, în general, agenții care licitează pentru oferirea serviciilor. Există posibilitatea că un task să fie compus din mai multe subtaskuri, caz în care contractorii devin manageri.

Distribuirea task-urilor cuprinde următorii pași:

- Identificare problema: Un agent decide ca are un task pe care nu-l poate indeplini singur
- Anunt: Agentul trimite catre toti agenții din sistem un anunt ce contine specificatiile task-ului de indeplinit, spre exemplu descrierea task-ului, constrangeri, termen limita etc.

- Licitare: Agenții care primesc anunțul, analizează cererea și decid dacă se vor implica în procesul de licitare, în funcție de capacitățile lor, de constrângeri și, în unele cazuri, de prețul task-ului.
- Alegerea castigatorului: Agentul care a inițiat anunțul alege castigatorul dintre agenții care au licitat.

În Fig. 4.11. sunt reprezentate specificatiile Contract Net Protocol dezvoltate de FIPA (*Foundation for Intelligent Physical Agents*). Acestea conțin următoarele etape:

- Inițiatorul trimite un anunț Call for Proposals (CFP) către m Participanți. Anunțul conține un termen limită pentru a răspunde la cerere.
- Fiecare participant care primește CFP, analizează cererea și licitează serviciile ce corespund cererii. Până la termenul limită, n Participanți au răspuns fie cu un mesaj de tip *refuse*, fie cu un mesaj *propose* care conține oferta.
- Inițiatorul analizează ofertele primite din mesajele de tip *propose* și alege cea mai bună ofertă. Participantul castigator va primi un mesaj de tip *accept-proposal*, iar restul Participanților vor primi un mesaj de tip *reject-proposal*.
- După finalizarea task-ului, Participantul trimite către Inițiator mesaj de tip *inform-done:inform* sau *inform-result:inform* dacă task-ul presupunea găsirea unei soluții sau mesaj *failure* însoțit de cauza eșecului, dacă agentul participant nu a reușit să finalizeze taskul.

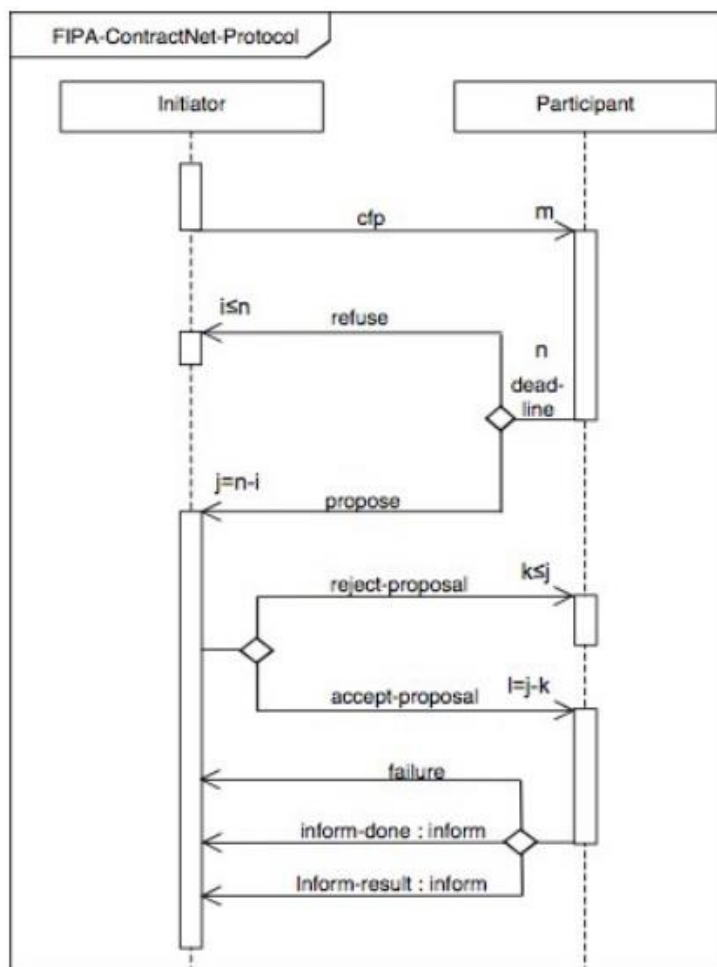


Figura 4.11. – Protocolul Contract Net dezvoltat de FIPA

Principalul avantaj al folosirii protocolului Contract Net ca mediu de comunicație într-o

rețea de agenți este că task-urile sunt atribuite dinamic și în mod echilibrat între agenți, deoarece agenții care au în sarcina alte task-uri pot alege să nu răspundă altor cereri.

Dezavantajele protocolului Contract Net sunt: infrastructură de comunicație nu este complet fiabilă, caz în care platforma multi-agent folosită pentru dezvoltarea sistemelor multi-agent trebuie să conțină un mecanism de prevenire și tratare a erorilor de comunicație. Folosirea Contract Net Protocol necesită, în unele situații, implementarea unui mecanism de detecție și rezolvare a conflictelor care pot apărea între agenți.

În cadrul studiului prezentat, comunicația dintre agenți este asigurată prin protocolul de comunicație Contract Net Protocol (CNP). Interacțiunea dintre agenți are loc în următoarele situații: (i) în etapă de inițializare a agenților Ordin, aceștia solicită și primesc de la agenții Produs informațiile legate de producție și produs (tipul de produs, lista și ordinea operațiilor necesare, cerințele de calitate și de utilizator etc), (ii) în etapă de alocare a operațiilor pe resurse, agenții Ordin interacționează cu agenții Resurse și (iii) în etapă de execuție a operațiilor, agenții Ordin solicită agenților Resurse să execute anumite operații, iar apoi, agenții Resursă notifică finalizarea execuției. Procesul de alocare a resurselor este detaliat mai jos și este reprezentat în Fig.4.12.

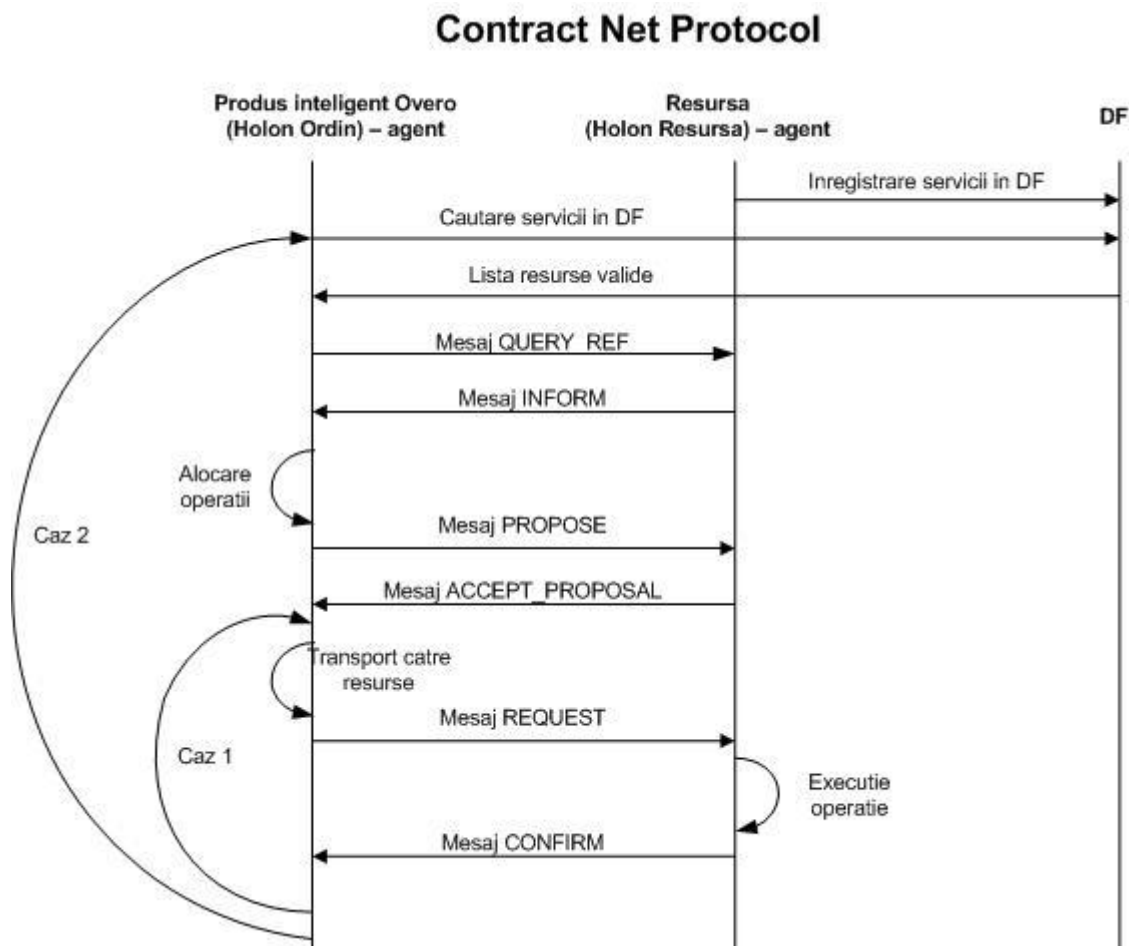


Fig. 4.12. Protocolul Contract Net folosit în procesul de alocare a operațiilor pe resurse

- Comunicația wireless

Fiecare agent de tip Ordin este localizat pe un Dispozitiv Inteligent Îmbarcat (IED – Intelligent Embedded Device) care are capabilități de comunicație wireless.

Dispozitivele inteligente sunt dotate cu antene de tip 802.11 și sunt conectate la un punct de acces(router) wireless localizat în interiorul celulei de fabricație care este conectat la rețeaua Ethernet din sistem.

Rețelele de tip WLAN (Wireless Local Area Network) permit conexiunea fără fir a două sau mai multe dispozitive la Internet, de regulă prin intermediul unui punct de acces. Astfel, dispozitivele rămân conectate la o rețea de tip WLAN atâta timp cât se află în aria de acoperire a rețelei respective. Aria de acoperire a rețelelor Wi-Fi este limitată, spre exemplu punctele de acces care folosesc standardul 802.11b sau 802.11g au o rază de acoperire de 35m în interior sau 100m în exterior.

Majoritatea rețelelor folosesc standardul IEEE 802.11, cunoscut și sub numele de Wi-Fi. IEEE 802.11 este un set de specificații pentru MAC (Media Access Control) și nivelul fizic privind implementarea rețelelor de tip WLAN.

Cel mai important avantaj al rețelelor wireless este mobilitatea. Lipsa cablului permite dispozitivelor mobile să se conecteze la rețeaua wireless din orice punct din aria de acoperire. Un alt avantaj este numărul mare de dispozitive care se pot conecta la același punct de acces.

Dezavantajele rețelelor wireless sunt securitatea și aria de acoperire. Pentru că o rețea wireless poate fi accesată de orice dispozitiv aflat în aria de acoperire, este necesară securizarea rețelelor care nu se doresc a fi publice. În unele situații, semnalul rețelei poate fi slab pentru că undele radio pot interfera cu alte unde sau cu obiecte din mediul înconjurător.

Utilizarea rețelelor wireless pentru implementarea sistemului multi-agent oferă mobilitate agenților de tip Ordin, deoarece oferă posibilitatea că agenții să fie executați direct pe dispozitivul inteligent.

Bibliografie

- Bussmann, S., & McFarlane, D. (1999). Rationales for Holonic Control Systems, Proceedings of *IMS99*(Leuven, Belgium).
- Dilts, D.M., Boyd, N.P., & Whorms, H.H. (1991). The Evolution of Control Architectures for Automated Manufacturing Systems. In *Journal of Manufacturing Systems*, Volume 10, Issue 1, Pages 79–93.
- Meyer, G. G., Främpling, K., Holmström, J.(2009). Intelligent Products: a survey. In *Computers in Industry*, 60, 137-148.
- Meyer, G. G., & Wortmann, H. (2010). Robust planning and control using intelligent product. In *W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, & E. David, et al. (Eds.), agent-mediated electronic commerce. Designing trading strategies and mechanisms for electronic markets. Lecture notes in business information processing (Vol. 59, pp. 163–177). Berlin Heidelberg: Springer.*
- McFarlane, D., Sarma, S., Chirn, J. L., Wong, C. Y., Ashton, K.(2002). The Intelligent Product in Manufacturing Control. In *Journal of EAIA*.
- McFarlane, D., Parlikad, A., Neely, A., Thorne, A. (2013). A framework for Distributed Intelligent Automation System Developments. In *Service Orientation in Holonic and Multi*

Agent Manufacturing and Robotics, Studies in Computational Intelligence, Volume 472, 2013, pp 313-326.

- McFarlane, D., Giannikas, V., Wong, C.Y., Harrison, M.(2013). Product intelligence in industrial control: Theory and practice. In *Annual Reviews in Control* 37, 69-88.
- Trentesaux, D., & Thomas, A. (2012). Product-Driven control: a State of the Art and Future Trends. In *INCOM 2012: 14th IFAC Symposium on Information Control Problems in Manufacturing*. Bucharest, Romania.
- Trentesaux, D., & Thomas, A. (2013). Product-Driven Control: Concept, Literature Review and Future Trends. In *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, Studies in Computational Intelligence* Volume 472, 2013, pp 135-150.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference Architecture for Holonic Manufacturing Systems: PROSA. In *Computers In Industry*, Vol. 37, No. 3, pp. 255 – 276.
- Wong, C.Y. , McFarlane, D., Zaharudin, A., Agarwal, V.(2002). The Intelligent Product Driven Supply Chain. In 2002 *IEEE International Conference on Systems, Man and Cybernetics*.
- Bellifemine F., Caire G., Greenwood D. (2007). *Developing Multi-Agent Systems with JADE*. John Wiley & Sons.
- Tutorial JADE <http://www.iro.umontreal.ca/~vaucher/Agents/Jade/JadePrimer.html>
- Configurare retea wireless modul Overo http://wiki.gumstix.org/index.php?title=Overo_Wifi