

# Developing Holonic Control Systems Using RFID Technology and IEC 61499 Models

Valentin VLAD, Adrian GRAUR, Cristina Elena TURCU

**Abstract**—This paper presents some concepts and solutions for developing holonic control systems. Some holonic models are introduced and validated within an RFID enabled demonstration environment.

RFID technology provides potential to increase the distribution and flexibility of a control system by attaching to raw materials or work in progress components RFID tags, containing all the necessary production information.

For the control application, the IEC 61499 architecture was chosen, given the advanced software technologies incorporated, like the encapsulation of functionality, component-based design, event-driven execution, and distribution.

**Index Terms**—distributed control, holonic systems, IEC 61499, RFID.

## I. INTRODUCERE

LITERATURA de specialitate conține mai multe lucrări științifice privind integrarea tehnologiei RFID în sisteme de fabricație holonice, atât la nivelul controlului din atelierul de producție, cât și la nivelul lanțurilor de distribuție a produselor.

Un exemplu semnificativ este reprezentat de [6], în care autorii prezintă o soluție holonică bazată pe tehnologia RFID pentru controlul proceselor de la nivelul atelierului de producție, în cadrul unor sisteme caracterizate de o varietate mare și un volum redus al produselor realizate. Ideea de bază are în vedere atașarea unui tag RFID pe fiecare componentă a unui produs, pe care vor fi înscrise în prealabil toate informațiile necesare finalizării produsului. Aceste tag-uri vor permite dirijarea și prelucrarea autonomă a componentelor de către dispozitivele din sistem, și în același timp adaptarea flexibilă a sistemului la schimbări prin actualizarea informațiilor pe care le conțin. Conceptul holonic este aplicat la nivelul produselor realizate prin considerarea fiecărei componente a unui produs ca reprezentând un holon, controlat de tag-ul RFID atașat.

O abordare oarecum similară este întâlnită în [8], în care autorii introduc conceptul de “produs inteligent”, care, de asemenea, are la bază aplicarea de tag-uri RFID pe componentele produselor realizate în cadrul unui sistem. Un “produs inteligent” este considerat a fi constituit dintr-o parte fizică, un set de informații caracteristice și o componentă agent, care asigură „comunicația produsului” cu elementele sistemului. El prezintă următoarele caracteristici:

- Deține o informație de identificare unică;
- Este capabil de a întreține o comunicație eficientă cu mediul în care se află;
- Poate reține sau stoca date despre sine;
- Își poate prezenta, prin intermediul unui limbaj, caracteristicile, cerințele de producție etc.;
- Este capabil să ia decizii cu privire la evoluția sa, sau să participe într-un astfel de proces.

În [11] se prezintă un exemplu de integrare a conceptelor holonice cu tehnologia RFID pentru urmărirea eficientă a produselor de-a lungul lanțurilor de aprovizionare. În cadrul lucrării autorii prezintă un metamodel de referință în care holonul “produs” este definit ca o entitate capabilă să îmbine partea fizică a produsului cu informația necesară urmării sale pe toată durata ciclului de viață.

Lucrarea de față este focalizată asupra îmbunătățirii flexibilității sistemelor de control de la nivelul atelierului de producție, prin îmbinarea într-un sistem holonic a avantajelor oferite de tehnologia RFID și, respectiv de modelele standardului IEC 61499. În următoarea secțiune sunt introduse câteva modele holonice, inspirate din societatea umană, împreună cu o soluție de aplicare a acestora în domeniul sistemelor de fabricație; secțiunea a III-a conține o descriere a standardului IEC 61499, ca și tehnologie folosită pentru implementarea acestor modele. Standardul IEC 61499 reprezintă un efort al Grupului de Lucru 6 al IEC (International Electrotechnical Commission) pentru dezvoltarea unei arhitecturi de control de nivel jos, agile și adaptive, care să constituie o bază pentru dezvoltarea sistemelor de fabricație holonice [15]. Secțiunea IV a lucrării descrie arhitectura sistemului fizic folosit pentru validarea modelelor propuse, secțiunea V prezintă un model holonic al sistemului de control dezvoltat, secțiunea VI conține un scenariu de lucru, iar în secțiunea VII sunt prezentate câteva aspecte privind modul de implementare al aplicației de control. Ultima secțiune este dedicată concluziilor și evidențierii contribuțiilor prezentate în cadrul lucrării.

## II. MODELE HOLONICE PENTRU DEZVOLTAREA SISTEMELOR DE CONTROL INDUSTRIALE

### A. Modele holonice sociale

Pornind de la principiile enunțate de Koestler [7] privind modul de organizare al sistemelor din lumea vie, în cadrul acestei secțiuni sunt prezentate câteva modele holonice, construite prin asemănare cu organismul uman și organizațiile

sociale.

În Fig. 1a este prezentat individul, ca și o entitate inteligentă de bază în cadrul unei organizații sociale. Privind în linii mari, acesta este constituit dintr-o componentă inteligentă, reprezentată de creierul sau mintea sa, și un set de componente integrate (membre, organe), care sunt aparent lipsite de inteligență, comparativ cu creierul. Aceste componente sunt foarte strâns conectate între ele și sunt subordonate componentei inteligente. Se poate observa însă și o anumită măsură de autonomie (cazul transplanturilor de organe) și o capacitate de reacție la acțiunile mediului (pe principiul stimul-răspuns), realizată în mod independent față de componenta inteligentă, creierul.

Privind acest mod de organizare prin prisma conceptelor holonice am putea considera organismul uman ca reprezentând un holon cu inteligență simplă, constituit dintr-o singură componentă inteligentă și un set de holoni oarecum lipsiți de inteligență, dar cu capacitate de integrare și subordonați componentei inteligente.

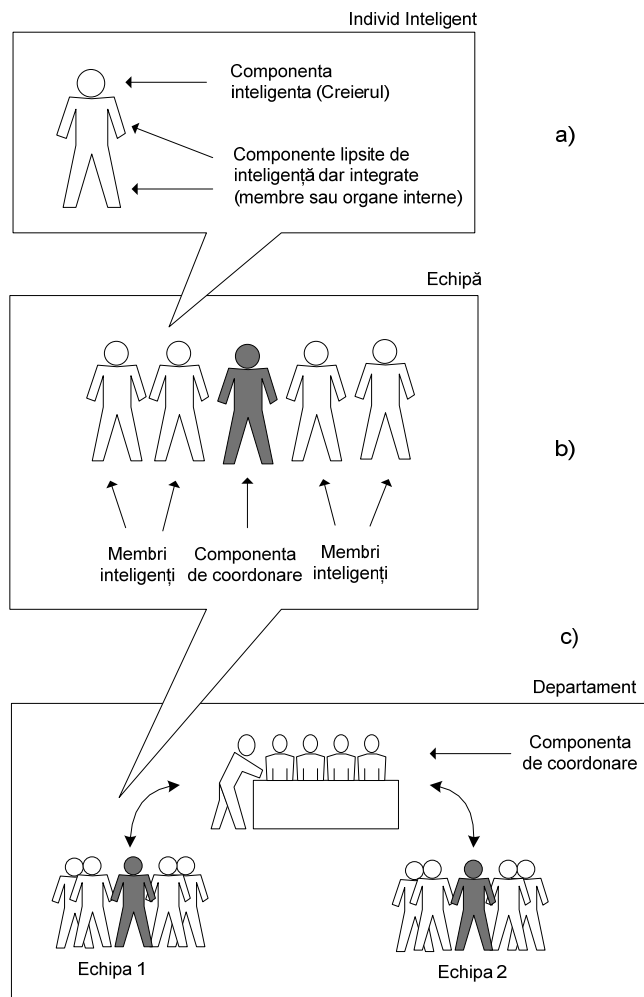


Fig. 1. Modele holonice sociale.

Analizând în continuare modul de organizarea al indivizilor în cazul în care se dorește realizarea unui anumit task, putem

observa principiul echipelor de lucru. O echipă este în general constituită din mai mulți indivizi, dintre care unuia îi este atribuită sarcina de coordonare și supraveghere a activităților desfășurate, așa după cum este ilustrat în Fig. 1b.

Echipa poate fi așadar privită ca un întreg constituit din mai mulți membri inteligenți, dar care manifestă un anumit grad de subordonare față de un membru special al echipei, cu atribute de conducere. Totuși, în cazul unei echipe, deciziile nu sunt luate doar de către componenta de coordonare, cum se întâmplă în cazul individului, ci sunt rezultatul unei interacțiuni dintre aceasta și ceilalți membri ai echipei.

Din punct de vedere al conceptelor holonice, echipa poate fi privită ca un holon cu inteligență complexă, constituit din mai mulți holoni cu inteligență simplă.

Dacă ar fi să considerăm în continuare cazul unui departament din cadrul unei organizații (Fig. 1c) am putea constata că acesta este constituit din mai multe echipe și un corp de conducere, având un mod de organizare și funcționare similar cu cel al echipei. Cu alte cuvinte, principiul echipei se poate observa pe toate nivelurile superioare acestuia, aplicat într-o manieră recursivă (echipe de echipe) și conducând la formarea de ierarhii.

Au fost identificate așadar trei tipuri de holoni, referiți în continuare ca și holoni de tipul I, II și III:

- *Holoni de tipul I* - holoni lipsiți de inteligență dar cu capacitate de integrare, foarte strâns legați fizic între ei, cu un grad redus de autonomie (manifestat printr-un comportament de tipul stimul-răspuns) și subordonați unei componente inteligente;
- *Holoni de tipul II* - holoni cu inteligență simplă, constituiți din mai mulți holoni de tipul I și o componentă inteligentă;
- *Holoni de tipul III* - holoni cu inteligență complexă, constituiți din mai mulți holoni de tipul II, dintre care unul deține sarcina de coordonare și supraveghere a activităților holonului, reprezentând totodată calea de comunicație cu alți holoni de același tip.

### B. Analogia cu domeniul sistemelor de fabricație

Arhitecturile holonice existente [12], [1] asociază în general fiecare componentă a unui sistem de fabricație (prodeuse, resurse, comenzi, activități logice) cu un *holon*. În cadrul acestei lucrări modelele holonice dezvoltate au fost aplicate doar resurselor, considerate ca actori inteligenți în cadrul procesului de producție.

Așa după cum se poate observa din Fig. 2, un holon de tipul I (notat cu H1) poate fi reprezentat de un dispozitiv de automatizare controlat în formă clasică, cum ar fi, de exemplu, un robot, un conveior controlat de un PLC sau o mașină cu comandă numerică. Un holon de tipul II (notat cu H2), poate încapsula unul sau mai multe astfel de dispozitive împreună cu o componentă inteligentă. Ca și exemplu pentru acest tip de holon se poate considera o celulă de asamblare constituită dintr-un robot de asamblare și un set de dispozitive de fixare și transport (Fig. 2), care conlucrează împreună sub

coordonarea unei componente software inteligente. În mod similar se poate considera cazul unui conveier inteligent obținut prin asocierea la un conveier clasic a unei componente inteligente care poate recepționa și executa comenzi de transportare a unor produse/componente spre anumite destinații.

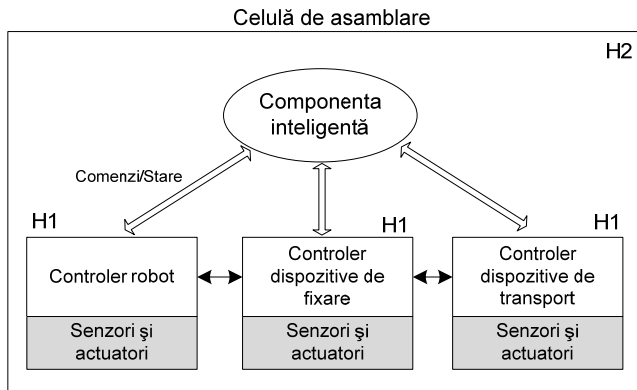


Fig. 2. Exemplu de holoni de tipul I și II.

Mai multe astfel de dispozitive inteligente (holoni de tipul II) pot forma o unitate de producție, coordonată de către o componentă software inteligentă de nivel superior și reprezentând un holon de tipul III.

### III. STANDARDUL IEC 61499 – INSTRUMENT PENTRU MODELAREA ȘI DEZVOLTAREA APLICAȚIILOR DE CONTROL INDUSTRIAL

IEC 61499 definește o arhitectură deschisă (eng. *open architecture*), modulară, destinată următoarei generații de sisteme din domeniul controlului distribuit și automatizării. Standardul încorporează tehnologii software avansate cum ar fi, încapsularea funcționalității, design modular, execuție controlată de evenimente și distribuție [1], [14].

Elementul de bază al arhitecturii îl reprezintă *blocul funcțional*, definit ca o unitate compozițională cu o interfață specifică și anumite elemente de funcționalitate, care pot fi implementate atât prin software, cât și prin hardware.

Alte elementele cheie definite în cadrul standardului sunt *aplicația*, *dispozitivul* și *resursa*. O aplicație este văzută ca o rețea de blocuri funcționale interconectate între ele, care definește în mod complet, dar abstract, funcționalitatea sistemului, fără a lua în considerare modul în care va fi distribuită la nivelul dispozitivelor de procesare. Un dispozitiv reprezintă o unitate de control care poate conține zero, una sau mai multe resurse, împreună cu interfețe de proces și interfețe de comunicație. O resursă poate fi privită ca un container pentru o rețea de blocuri funcționale și corespunde unui procesor pe care rulează o parte din aplicația distribuită.

#### A. Blocul funcțional

Interfața unui bloc funcțional (Fig. 3) este definită ca o listă de intrări de evenimente, intrări de date și socket-uri pentru blocuri de tip *adapter*, ieșiri de evenimente, ieșiri de date și “fișe” (eng. *plug*) de adaptare.

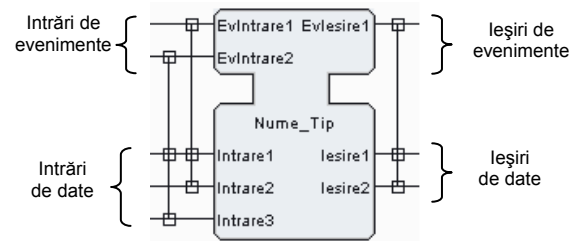


Fig. 3. Interfața unui bloc funcțional.

Reprezentarea grafică a unui bloc funcțional poate fi văzută ca fiind compusă din două componente, denumite “cap” (partea superioară) și “trup” (partea inferioară). *Capul* conține intrările și ieșirile de evenimente, în timp ce intrările și ieșirile de date sunt conectate la *trupul* blocului funcțional.

IEC 61499 permite definirea a diferite tipuri de blocuri funcționale, într-o manieră similară standardului IEC 61131 sau programării orientată obiect. Odată definite, tipurile de blocuri funcționale pot fi stocate în cadrul unei biblioteci și apoi instanțiate în cadrul aplicațiilor ori de câte ori este nevoie.

Standardul definește mai multe categorii de blocuri funcționale: de bază, compuse și de interfațare a serviciilor. Diferența dintre ele constă în modul în care este descrisă structura internă și funcționalitatea încapsulată, definirea interfeței realizându-se în mod similar pentru toate tipurile.

#### B. Blocuri funcționale de bază

Un bloc funcțional de bază reprezintă o structură software cu rolul de a executa o funcție elementară din cadrul aplicației de control, cum ar fi, citirea unui senzor sau setarea stării unui actuator. După cum se poate observa și din Fig. 4, un astfel de bloc conține *variabile interne*, unul sau mai mulți *algoritmi* și o *diagramă de control a execuției* (eng. *execution control chart - ECC*).

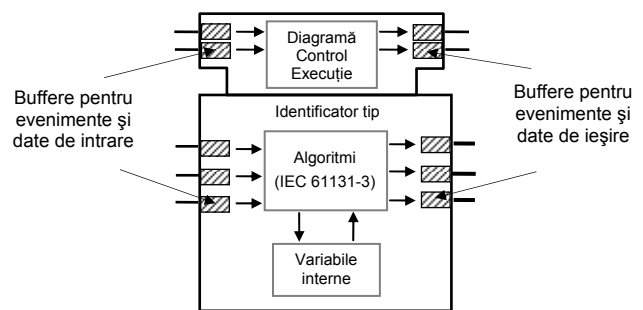


Fig. 4. Bloc funcțional de bază.

#### C. Blocuri funcționale compuse

În contrast cu blocurile funcționale de bază, funcționalitatea încapsulată într-un bloc funcțional compus este determinată de o rețea de blocuri funcționale interconectate între ele, așa după cum este ilustrat în Fig. 5. Blocurile care formează rețeaua pot fi de bază, de interfațare a serviciilor sau chiar compuse, permițând construirea de aplicații organizate ierarhic.

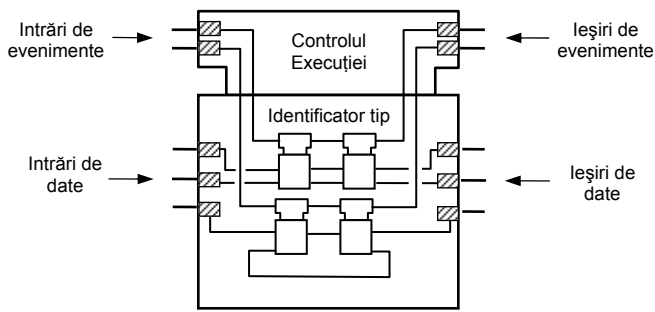


Fig. 5. Bloc funcțional compus.

#### D. Blocuri funcționale de interfațare a serviciilor

Blocurile funcționale de interfațare a serviciilor au rolul de a abstractiza serviciile pe care o anumită resursă le oferă aplicației de control (cum ar fi citirea valorii unui senzor sau setarea stării unui actuator). Implementarea lor implică cunoașterea detaliilor hardware a unei anumite platforme și trebuie să fie asigurată de către producătorii de echipamente, în mod similar *driverelor* corespunzătoare componentelor din cadrul unui PC.

Un loc aparte în cadrul blocurilor funcționale de interfațare a serviciilor îl au cele destinate comunicației în cadrul sistemului distribuit, denumite Blocuri Funcționale de Interfațare a Comunicației.

Standardul definește în mod explicit două modele generice de comunicație: PUBLISH/SUBSCRIBE, pentru tranzații uni-direcționale, și CLIENT/SERVER pentru comunicație bi-direcțională. În cazul modelului PUBLISH/SUBSCRIBE comunicația este “cu difuzare”, astfel încât mai multe blocuri de tipul SUBSCRIBE pot recepționa date de la același bloc PUBLISH și, de asemenea, mai multe blocuri PUBLISH pot trimite date către același SUBSCRIBER.

În cadrul acestei lucrări s-a ales modelul PUBLISH/SUBSCRIBE pentru comunicația inter-holonică, iar pentru comunicația intra-holonică (dintre componenta inteligentă și controlerele dispozitivelor fizice) s-a utilizat modelul CLIENT/SERVER.

#### E. Soluții pentru dezvoltarea aplicațiilor conforme cu IEC 61499

Function Block Development Kit (FBDK) reprezintă în acest moment unul din cele mai evolute medii pentru dezvoltarea de aplicații distribuite, conforme cu specificațiile standardului IEC 61499. Acest mediu a fost dezvoltat inițial în cadrul companiei Rockwell Automation, servind ca un instrument pentru validarea diferitelor specificații ale standardului IEC 61499, aflat atunci în curs de dezvoltare. În prezent mediul FBDK este întreținut și distribuit de către organizația HOLOBLOC Inc. (USA), putând fi descărcat gratuit de pe site-ul [www.holobloc.com](http://www.holobloc.com). Se pot menționa și alte soluții similare dezvoltate în diverse centre de cercetare, cum ar fi:

- *CORFU Engineering Support System* [1] – dezvoltat în cadrul Universității din Patras, Grecia, în cadrul căruia

se încearcă o extindere a modelului IEC 61499 pentru specificarea cerințelor unui sistem cu ajutorul UML (Unified Modelling Language) [10].

- *ISaGRAF* – reprezintă primul mediu de dezvoltare comercial pentru programarea PLC-urilor care a inclus în 2005, pe lângă specificațiile standardului IEC 61131, suport pentru dezvoltarea de aplicații conforme cu standardul IEC 61499 [5].
- *O<sup>3</sup>neida Fbench* – reprezintă un proiect “open source” dezvoltat în cadrul Universității din Auckland (Noua Zeelandă), în colaborare cu grupul O<sup>3</sup>neida. Acest mediu este foarte similar cu FBDK, și se dorește a fi un punct de colaborare dintre diverși cercetători din domeniu [1].

Pentru dezvoltarea aplicațiilor practice din cadrul acestei lucrări s-a folosit mediul FBDK.

FBDK include două componente principale:

- Un *mediu integrat de dezvoltare*, denumit Editor (Fig. 6), care asigură suport pentru dezvoltarea grafică de blocuri funcționale și sisteme, precum și pentru translatarea acestora în clase Java.
- Un *mediu de rulare*, denumit FBRT (Function Block Run-Time), care permite rularea aplicațiilor dezvoltate în cadrul Editor-ului pe orice sistem care dispune de Mașina Virtuală Java. Structura mediului de rulare este ilustrată în Fig. 7.

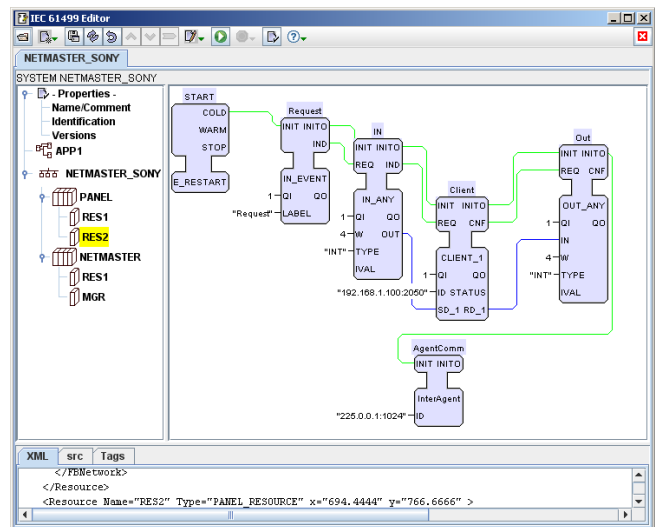


Fig. 6. Mediul integrat de dezvoltare al FBDK.

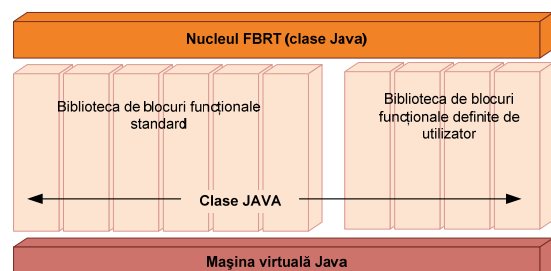


Fig. 7. Structura mediului de rulare FBRT.

Comunicația dintre componenta Editor și mediul de rulare FBRT se realizează prin intermediul unui protocol de management de tipul XML [14].

#### IV. ARHITECTURA SISTEMULUI FIZIC FOLOSIT PENTRU VALIDAREA MODELELOR HOLONICE PROPUSE

Mediul fizic pentru validarea modelelor propuse este construit în jurul unui conveier în buclă închisă, cu trei fluxuri, produs de firma Bosch Rexroth (Fig. 8). Pe fiecare flux al conveierului, în dreptul unei bariere, este plasat câte un reader RFID (de tipul Feig ID CPR.02), conectat la un controler embedded cu rol în simularea unei stații de prelucrare.

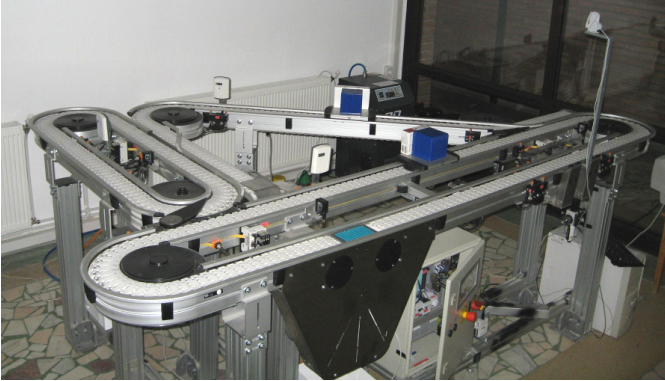


Fig. 8. Sistemul fizic pentru testarea aplicației de control

Conveierul Bosch este controlat de un PLC IndraControl L40, la care sunt conectați, printr-o magistrală AS-i sau prin intrări/ieșiri digitale directe, toți senzorii și elementele de acționare implicate în procesul de transport. Acestea sunt reprezentate de senzori de proximitate, cititoare în radio frecvență, comutatoare de traseu, bariere și contactoare pentru motoarele de antrenare a benzilor. Fig. 9 conține o reprezentare schematică simplificată a sistemului fizic.

Din punct de vedere al comunicației PLC-ul IndraControl L40 este echipat cu un port serial RS-232 și o interfață Ethernet (TCP/IP), care îi permite conectarea într-o rețea LAN (Local Area Network).

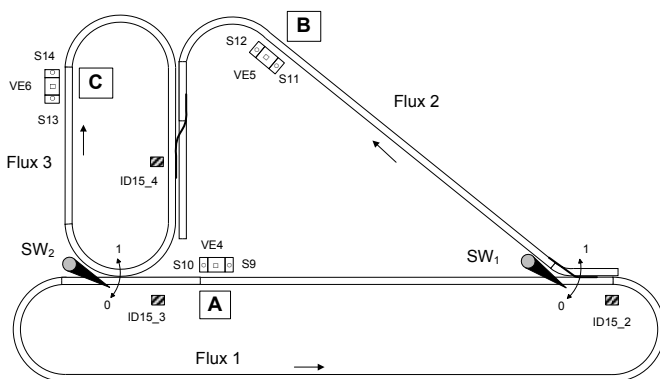


Fig. 9. Reprezentarea schematică simplificată a sistemului fizic

Cele trei “stații de prelucrare”, notate cu A, B și C (în Fig. 9), sunt plasate în dreptul barierei VE4, VE5, respectiv

VE6. Fiecare barieră este însoțită de doi senzori de proximitate, așezați unul în față și celălalt în spatele barierei, care au rolul de anunța sosirea unui palet, respectiv de a confirma trecerea acestuia de barieră. Comutarea între fluxuri se realizează cu ajutorul a două comutatoare de traseu, notate cu  $SW_1$  și  $SW_2$ .

Paleții transportați pe benzile conveierului includ un cip RFID de tipul Bosch Rexroth ID15/MDT, care asigură o identificare unică în cadrul sistemului. Pe fluxurile conveierului, înaintea comutatoarelor de traseu, sunt amplasate dispozitive de citire/scriere de tipul ID15/SLK, care permite citirea/modificarea datelor din cipurile incluse în paleți. În Fig. 9 aceste dispozitive sunt notate cu ID15\_2, ID15\_3, respectiv ID15\_4.

Pe lângă cip-urile RFID incluse în paleți, fiecare componentă transportată de un palet are atașat un tag RFID care operează la frecvența de 13,56 MHz și care poate fi citit/scriș numai de reader-ele RFID conectate la controlerele “stațiilor de prelucrare”. Aceste tag-uri conțin operațiile care trebuie executate asupra componentei, împreună cu numărul de identificare al paletului care o transportă.

În acest sens a fost adăugată în cadrul sistemului o stație similară “stațiilor de prelucrare” cu rol în formatarea inițială a tag-urilor atașate componentelor, funcție de comenzile primite și de paleții pe care sunt transportate componentele.

Controlerele embedded pentru simularea stațiilor de prelucrare sunt echipate cu microcontrolere din familia ARM și pot realiza operații de citire sau scriere asupra unui tag RFID prin intermediul reader-ului asociat. Controlerele prezintă o interfață de comunicație RS-485 care le permite conectarea în rețea în scopul comunicării cu componentele software de coordonare ale sistemului.

#### V. MODELUL HOLONIC AL SISTEMULUI DE CONTROL

“Stațiile de prelucrare” prezentate, stația pentru formatarea tag-urilor și conveierul care asigură transportarea paleților sunt controlate într-o formă clasică. Acest lucru înseamnă că este imposibilă conectarea lor directă într-un sistem inteligent distribuit. Conform conceptelor holonice definite anterior aceste dispozitive reprezintă holoni de tipul I, cărora trebuie să li se asocieze câte o componentă inteligentă pentru a deveni entități inteligente (adică holoni de tipul II) în cadrul sistemului. Aceste entități pot conlucra apoi sub coordonarea unei componente inteligente superioare, formând un holon de tipul III (H3). Fig. 10 ilustrează ideea de holonificare și intercomunicație a dispozitivelor din sistem.

Arhitectura holonilor pentru “stațiile de prelucrare”, respectiv pentru conveier (Fig. 11) este constituită din controlerul local al dispozitivului fizic, o interfață independentă de hardware și o componentă agent, care asigură integrarea dispozitivului în sistemul de control inteligent. În cazul holonilor pentru “stațiile de prelucrare” componenta pentru interfața independentă de hardware este localizată pe un dispozitiv embedded, privit ca un controler adițional care asigură interfațarea fizică și logică a controlerului stației.

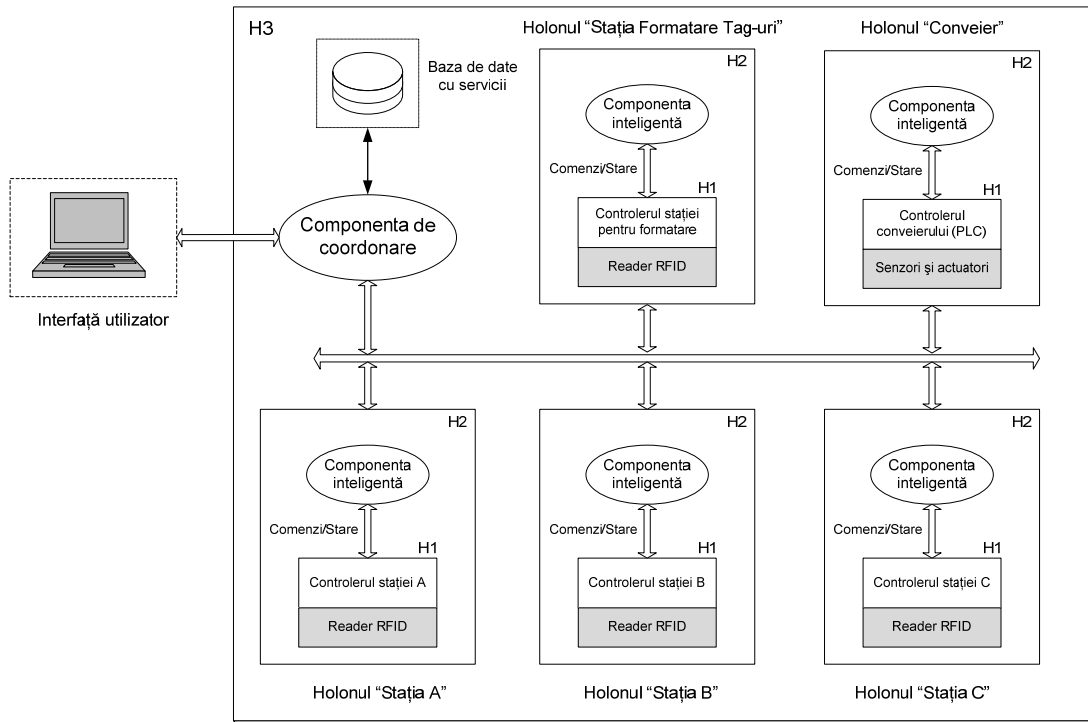


Fig. 10. Holonificarea dispozitivelor din sistem.

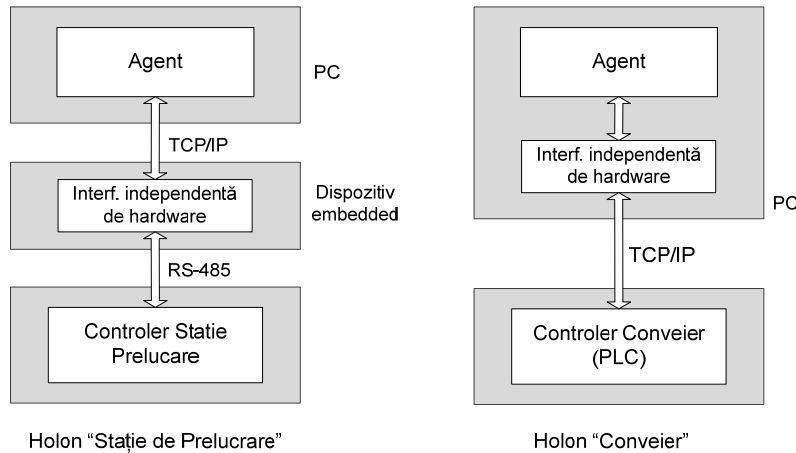


Fig. 11. Arhitectura holonilor pentru "stațiile de prelucrare" și conveier.

În cazul holonului corespunzător conveierului componenta pentru interfața independentă de hardware prezintă un grad de complexitate mai ridicat și rulează pe un sistem de calcul de tip PC, cu resurse superioare dispozitivului embedded.

## VI. SCENARIU DE LUCRU

Prin scenariul de lucru prezentat în cadrul acestei subsecțiuni se încearcă punerea în evidență a avantajelor oferite de îmbinarea conceptelor holonice cu tehnologia RFID în realizarea sistemelor de control inteligente distribuite.

Așa după cum s-a menționat deja, asocierea de tag-uri RFID componentelor produselor și înscrierea în ele a

informațiilor necesare prelucrării acestora poate conduce la așa numita "fabricație condusă de produs" (eng. *product-driven manufacturing*), în care produsul deține un rol activ în procesul realizării sale. În cadrul aplicației implementate acest lucru se rezumă la două aspecte:

- O componentă semifabricat poate fi dirijată automat către stația de prelucrare destinație, pe baza adresei acesteia, înscrisă în tag-ul RFID atașat componentei.
- O stație de prelucrare poate executa în mod autonom anumite operații asupra unei componente, tipul și caracteristicile acesteia fiind înscrisse în tag-ul RFID atașat.

Ca și exemplu se poate considera o comandă de realizare a

unui produs de tipul  $P_2$ , constând din 3 servicii, toate realizate asupra aceleiași componente semifabricat:

$$P_2 = \{S_4, S_5, S_6\} \quad (1)$$

Comanda este introdusă prin intermediul interfeței utilizator și preluată de agentul de coordonare a unității de producție. Acesta trebuie să consulte baza de date a serviciilor pentru a determina serviciile care compun produsul  $P_1$  și “stațiile de prelucrare” capabile să le execute. Considerând tabela cu servicii ca fiind cea din Fig. 12, componenta va trebui să viziteze succesiv stațiile A, B și C, care au adresele 1, 2, respectiv 3.

SERVICES : Table			
	Address	Service	Device Descript
	5	S1	Assembly
	5	S2	Assembly
	5	S3	Assembly
	1	S4	Statia A
	2	S5	Statia B
	3	S6	Statia C
	*		

Fig. 12. Exemplu de tabelă cu serviciile dispozitivelor din sistem.

Ca urmare, componenta de coordonare va cere stației de formare a tag-urilor crearea unui tag nou (din punct de vedere logic) care să conțină serviciile ce trebuie executate asupra componentei, adresele stațiilor care le vor realiza și identificatorul paletului care va transporta componenta. Pentru exemplul considerat vom presupune că identificatorul paletului are valoarea 20. Tag-ul RFID astfel creat va fi atașat componentei semifabricat, iar agentul de coordonare va emite către conveier o cerere de transportare a paletului cu ID-ul 20 către stația cu adresa 1.

Dirijarea paletului către o anumită stație se realizează pe baza a două tabele de rutare (una pentru comutatorul  $SW_1$  și cealaltă pentru comutatorul  $SW_2$ ), construite în faza de configurare inițială a sistemului [13] și care conțin o asociere dintre adresa fiecărei stații și poziția comutatorului de traseu pentru dirijarea paletului spre acea stație pe drumul cel mai scurt.

Paletul ajuns în dreptul unei “stații de prelucrare” (reader RFID) va fi blocat de bariera asociată acesteia, așa încât stația poate citi sau modifica conținutul tag-ului RFID atașat componentei. Dacă stația constată că ea reprezintă destinația paletului va marca în tag-ul RFID serviciul cerut ca fiind realizat. Va citi apoi adresa următoarei destinații a paletului și va emite către conveier o cerere de transportare a paletului către aceasta. Dacă serviciul tocmai “realizat” este ultimul din lista conținută de tag-ul RFID, stația va cere conveierului dirijarea paletului către un punct de ieșire.

## VII. ASPECTE PRIVIND IMPLEMENTAREA SISTEMULUI DE CONTROL

Controlerele fizice pe care rulează aplicația distribuită sunt prezentate în Fig. 14 și constau în: PLC-ul pentru controlul conveierului (IndraControl L40), controlerele pentru simularea stațiilor de prelucrare, controlerul stației pentru

formatarea tag-urilor RFID, un dispozitiv embedded Elsist Netmaster [9] și un sistem de calcul de tip PC.

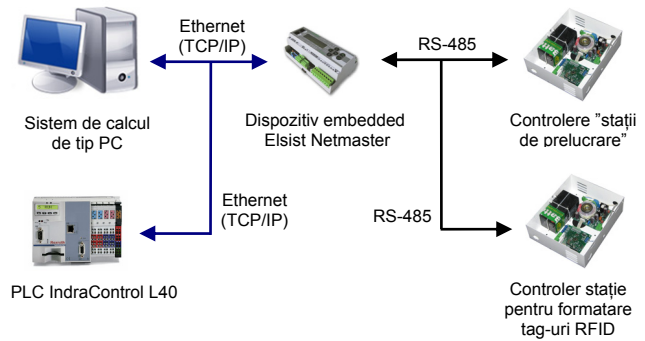


Fig. 14. Controlerele fizice pe care rulează aplicația distribuită.

Componenta software pentru interfața independentă de hardware a controlerului conveierului rulează pe sistemul de calcul de tip PC, iar cele pentru controlerele “stațiilor de prelucrare”, respectiv pentru stația care asigură formatarea tag-urilor RFID rulează pe dispozitivul embedded Elsist Netmaster. Componentele *agent* pentru toate cele trei tipuri de controlere rulează pe sistemul de tip PC.

Componentele *agent* și interfețele independente de hardware sunt dezvoltate conform specificațiilor IEC 61499, cu ajutorul mediului FBDK. Aplicația din PLC-ul IndraControl L40 este dezvoltată în cadrul mediului Bosch IndraLogic, conform IEC 61131-3, iar aplicațiile din controlerele stațiilor de prelucrare, respectiv a stației pentru formatarea tag-urilor RFID, sunt dezvoltate în jurul sistemului de operare ARTX produs de firma Keil.

### A. Aplicația dezvoltată în mediul FBDK

Aplicația construită în FBDK este constituită din două dispozitive: PANEL, care rulează pe sistemul de calcul de tip PC, și NETMASTER, care rulează pe dispozitivul embedded Elsist Netmaster (Fig. 15).

Resursa *Agents* din cadrul dispozitivului PANEL (Fig. 15) conține blocuri funcționale pentru agenții celor trei “stații de prelucrare”, pentru stația care se ocupă de formatarea tag-urilor RFID, pentru conveier și pentru Agentul de Coordonare.

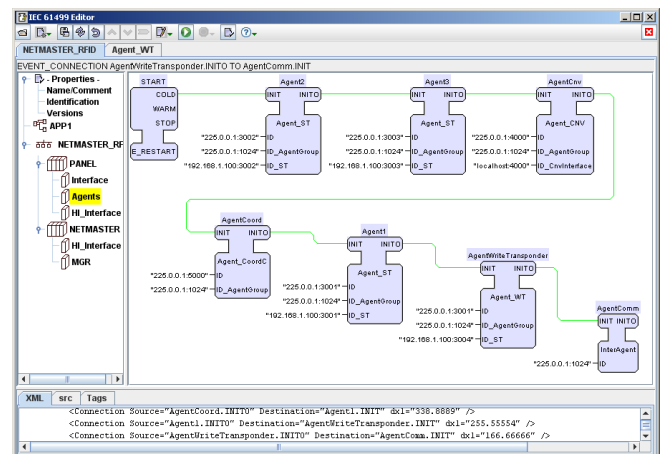


Fig. 15. Aplicația de control dezvoltată în FBDK.

Resursa *HI\_Interface* din cadrul dispozitivului PANEL conține componenta pentru interfața independentă de hardware a controlerului conveierului, implementată în forma unui bloc funcțional compus. Un bloc semnificativ din cadrul acestei interfețe este reprezentat de blocul de interfațare a serviciilor de tipul Cnv\_IO, încapsulat în blocul funcțional de tipul Cnv\_IO\_C (Fig. 16). Acest bloc conține codul Java pentru comunicația TCP/IP cu aplicația din PLC-ul conveierului și furnizează aplicației valorile senzorilor care echează conveierul, respectiv permite acesteia setarea stării elementelor de acționare (pentru acționarea barierelor).

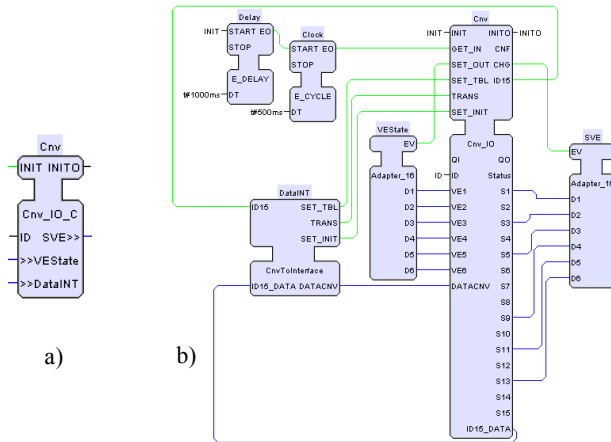


Fig. 16. Interfața (a) și conținutul (b) blocului funcțional Cnv\_IO\_C.

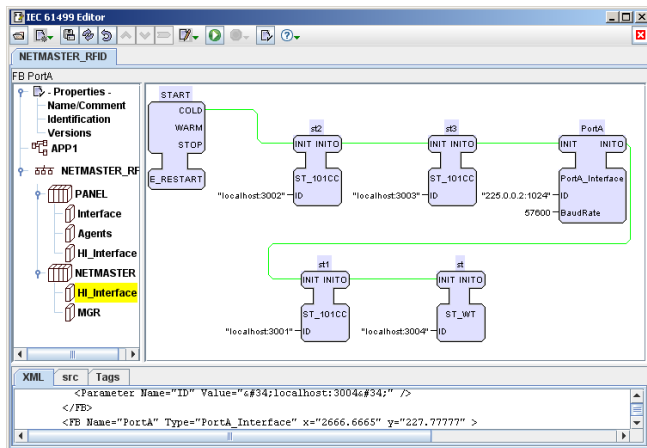


Fig. 17. Resursa *HI\_Interface* din cadrul dispozitivului NETMASTER.

Resursa *HI\_Interface* din cadrul dispozitivului NETMASTER (Fig. 17) conține interfețele independente de hardware pentru “stațiile de prelucrare” și stația pentru formatarea tag-urilor RFID. Comunicația dintre aceste interfețe și controlerul stațiilor se realizează printr-un port serial al dispozitivului Elsist Netmaster. Interfața de comunicație corespunzătoare portului serial este implementată în forma unui bloc funcțional compus și inclusă tot în cadrul acestei resurse.

## VIII. CONCLUZII

În cadrul acestei lucrări a fost prezentată o soluție de realizare a unui sistem de control inteligent distribuit prin îmbinarea conceptelor holonice cu modelele standardului IEC 61499 și avantajele oferite de tehnologia RFID. Contribuțiile, atât de ordin teoretic cât și practic, includ:

- Identificarea unor modele holonice sociale și definirea a trei tipuri de holoni (holoni de tipul I, II, și III) ca și unități componențiale în cadrul acestor modele;
- Prezentarea unei modalități de aplicare a acestor modele în domeniul sistemelor de fabricație;
- Definirea și implementarea unei soluții practice pentru controlul transportării și prelucrării autonome a componentelor semifabricat, în care “produsul” deține un rol activ prin informațiile înscrise în tag-ul RFID atașat;
- Proiectarea și dezvoltarea aplicației de control, de nivel înalt, în conformitate cu specificațiile standardului IEC 61499 prin extinderea aplicabilității standardului spre componente de control de nivel înalt (cum ar fi cazul unor agenți inteligenți).

## REFERENCES

- [1] Chirn, J., McFarlane, D.C. “A Holonic Component-Based Approach to Reconfigurable Manufacturing Control Architecture”, 11th International Workshop on Database and Expert Systems Applications (DEXA'00), 2000, pp. 219-223.
- [2] CORFU Engineering Support System. Web site: <http://seg.ece.upatras.gr/corfu/dev/index.htm>
- [3] FBench – Mediu de dezvoltare pentru aplicații conforme cu IEC 61499. Web site: <http://www.ele.auckland.ac.nz/~vyatkin/fbench/index.html>, 2009.
- [4] Function Blocks for Industrial Process Measurement and Control Systems, Standard, International Electrotechnical Commission, Geneva, 2005.
- [5] ISaGRAF. Web site: <http://www.isagraf.com>.
- [6] Kamioka, K., Kamioka, E., Yamada, S. “An RF-ID driven holonic control scheme for production control systems”, Proceedings of the 2007 International Conference on Intelligent Pervasive Computing, pp. 509-514, 2007.
- [7] Koestler, A., Smythies, J.R., Beyond Reductionism, Hutchinson, 1972.
- [8] McFarlane, D, Sarma, S., Chirn, J.L., Wong, C.Y., Ashton, K. “The Intelligent Product in Manufacturing Control”, Journal of EAIA, Iulie 2002.
- [9] Netmaster, Elsist - web site. <http://www.elsist.it>.
- [10] Object Management Group. Web site: [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm)
- [11] Terzi, S., Panetto, H., Morel, G., Garetti, M. “A holonic metamodel for product traceability in PLM”, International Journal of Product Lifecycle Management, vol. 2, 2007 pp. 253-289.
- [12] Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. Peeters, P. „Reference architecture for holonic manufacturing systems: PROSA”, Computers in Industry, Vol. 37, 1998.
- [13] Vlad, V., Graur, A., Turcu, C.E., Popa, C. “Enhancing the Flexibility of Manufacturing Systems Using the RFID Technology”, 2009 International Conference on Advanced Information Networking and Applications Workshops, pp. 630-635, 2009.
- [14] Vyatkin, V. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, Instrumentation Society of America, USA, 2007.
- [15] Zoitl, A. Real-Time Execution for IEC 61499, ISA & O3neida, 2009.