

Address Space Modeling for a New Generation of Middleware - A Review

Simona-Anda GHERASIM, Vasile Gheorghită GĂITAN, Daniel SIMION
 “Ștefan cel Mare” University of Suceava
 sgherasim@stud.usv.ro

Abstract—The data exchange in automated industry is one of the most difficult requirements that must achieve a middleware application. Researchers have tried different solutions to achieve this exchange of data in order to be more reliable. Starting from applications based on COM / DCOM, the most important specifications for automated data exchange industry, has been provided by the OPC Foundation. These specifications are widely accepted in industry and are already implemented by any system of automated industry.

Index Terms—address space, middleware application, informational model, hierarchical types.

I. INTRODUCERE

MOTIVAȚIA - Schimbul de date în industria automatizată este una din cele mai dificile cerințe pe care trebuie să le realizeze o aplicație de tip middleware. În acest sens, s-au căutat diferite soluții pentru ca realizarea acestui schimb de date să se facă într-un mod cât mai fiabil.

Plecând de la aplicațiile bazate pe COM/DCOM, cele mai importante specificații pentru schimbul de date din industria automatizată, au fost furnizate de către OPC Foundation. Astfel, cele mai proeminente specificații sunt OPC Data Access (DA), OPC Alarm & Events (A&E), OPC Historical Data Access (HDA), care sunt acceptate pe scară largă în industrie, fiind puse deja în aplicare de către orice sistem din industria automatizată.

Pe lângă aceste specificații, OPC Foundation a lansat o nouă generație de specificații, și anume OPC Unified Architecture (UA), OPC eXpress Interface (OPC XI) care realizează o schimbare de tehnologie, o trecere de la vechea tehnologie bazată pe COM/DCOM la o arhitectură orientată spre servicii, furnizând datele într-o manieră independentă de platformă, prin intermediul serviciilor web sau prin optimizările proprii bazate pe protocolul TCP. OPC UA unifică specificațiile anterioare într-un spațiu de adrese unic, capabil să se ocupe cu datele actuale, alarme și evenimente, precum și cu istoricul datelor actuale și istoricul evenimentelor.

O îmbunătățire remarcabilă a OPC Unified Architecture

este modelul spațiului de adrese, prin care furnizorii pot expune un model de informații bogate și extensibile utilizând tehnici orientate pe obiecte. OPC UA scalează foarte bine începând de la dispozitive inteligente, controlere, sisteme DCS (Distributed Control System), sisteme SCADA și până la sisteme MES și ERP. De asemenea, scalează foarte bine prin capacitatea sa de a furniza informații. În partea inferioară, poate fi folosit un model similar cu cel al OPC-ului clasic, furnizând doar informații de bază, în timp ce, în partea superioară, pot fi descrise modele extrem de sofisticate, oferind o cantitate mare de metadate, inclusiv ierarhii complexe de tipuri.

Aici apare un mare interes în capacitatea de modelare avansată în multe domenii, existând de asemenea, inițiative de a standardiza modelele informaționale bazate pe OPC UA. Un exemplu în acest sens, îl reprezintă FDI (Field Device Integration), unde o descriere comună a dispozitivelor de câmp și activitățile comune sunt direcționate spre MIMOSA (întreținerea informațiilor - ERP și cele amintite mai sus), S95 (producerea informațiilor - MES), și PLCopen (controlul industrial).

Caietul de sarcini al OPC UA, definește, în primul rând, modul în care trebuie făcut un lucru și într-o mai mică măsură explică de ce a fost proiectat în acel mod. Aceasta este maniera prin care, specificația trebuie să fie scrisă pentru a fi aplicată pe scară largă, de către cât mai mulți dezvoltatori și pentru a garanta interoperabilitatea între diferite aplicații. Ca urmare, specificația OPC UA este foarte greu de înțeles de cineva nou în domeniu. De asemenea, acest lucru este adevărat și pentru alte specificații precum SQL sau UML. [1] [2] [3]

II. SPAȚIUL DE ADRESE

Spațiul de adrese reprezintă cantitatea de memorie alocată pentru toate adresele posibile unei entități de calcul, cum ar fi: un dispozitiv, un fișier, un server sau un computer din rețea. Astfel conceptul de spațiu de adrese se poate referi la o serie de adrese, fie fizice sau virtuale, accesibile unui procesor sau rezervate unui proces. Ca identificatori unici ai singurelor entități, fiecare adresă specifică locația unei entități (unitatea de memorie care poate fi adresată separat). Pe un calculator, fiecărui dispozitiv sau proces, îi este alocat un spațiu de adresă, care reprezintă o parte din spațiul de adrese alocat procesorului.

Cu alte cuvinte, spațiul de adrese reprezintă gama de adrese virtuale care poate fi atribuită:

Simona-Anda GHERASIM – Universitatea Ștefan Cel Mare Suceava, str.Universitatii nr.13, RO-720229 Suceava (e-mail: sgherasim@stud.usv.ro)

Vasile Gheorghită GĂITAN– Universitatea Ștefan Cel Mare Suceava, str.Universitatii nr.13, RO-720229 Suceava (e-mail: gaitan@eed.usv.ro)

Daniel SIMION - Universitatea Ștefan Cel Mare Suceava, str.Universitatii nr.13, RO-720229 Suceava.

- unui utilizator, gazdă din rețea,
- unui dispozitiv periferic, sector de disc,
- unei celule de memorie sau altă entitate logică sau fizică,
- sau separat unor programe în rulare.

Spațiul de adrese al unui proces este mulțimea locațiilor alocate acestuia. [4]

III. SPAȚIUL DE ADRESE LA OPC-UL CLASIC

OPC - Object Linking and Embedding for Process Control este binecunoscutul standard specific schimbului de date dintre diferitele sisteme industriale automatizate din producție și industria proceselor.

OPC folosește o abordare de tip client-server pentru schimbul de informații. Un server OPC încapsulează sursele informațiilor unui proces și le face valide pentru interfețele definite. Clientul OPC se conectează la serverul OPC și poate accesa sau prelucra datele oferite. Aplicațiile care prelucrează și produc date, pot fi: fie client, fie server.

Interfețele OPC-ului clasic sunt bazate pe tehnologiile COM și DCOM de la Microsoft.

Avantajul acestei abordări a fost reprezentat de reducerea caietului sarcinilor de lucru la definirea diferitelor funcții API pentru diferite nevoi specializate, fără a fi nevoie să se definească un protocol de rețea sau un mecanism de inter-comunicare. COM și DCOM furnizează un mecanism transparent pentru ca un client să poată apela metodele de pe un obiect COM dintr-un server care rulează în același proces, pe un alt proces sau pe un alt nod de rețea.

În cadrul Specificației OPC Data Access - obiectele server oferă metode pentru navigarea prin ierarhia spațiului de adrese pentru a identifica obiectele și proprietățile acestora, precum tipul de dată și drepturile de acces.

Pentru a accesa datele, clientul grupează elementele OPC cu setări identice, precum timpul de actualizare, într-un obiect OPCGroup. Figura 1 ilustrează diferite obiecte create de clientul OPC în server.

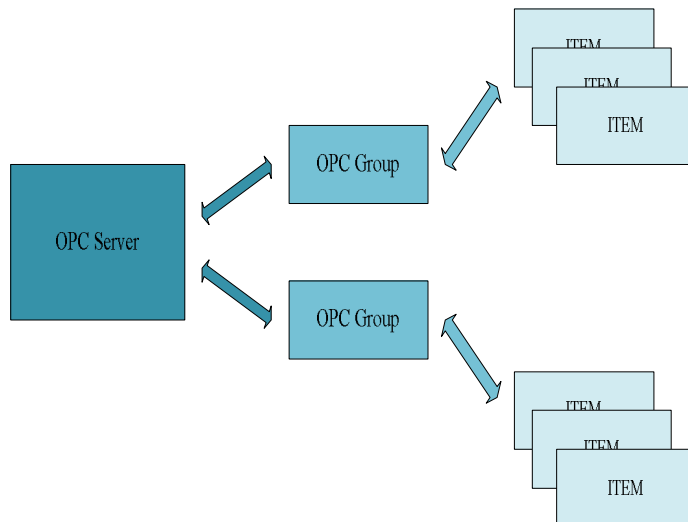


Fig. 1. Obiecte create de către clientul OPC în interiorul serverului pentru a accesa datele

Când sunt adăugate la un grup, elementele pot fi citite sau scrise de către client. Totuși, calea preferată pentru ciclul de citire al datelor de către client este monitorizarea schimbărilor valorilor în server. Clientul definește o rată de actualizare în cadrul grupului care conține elementele de interes. Rata de actualizare este folosită în cadrul serverului pentru a verifica ciclic valorile care se schimbă. După fiecare ciclu, serverul îi trimite clientului doar valorile care s-au schimbat.

OPC furnizează datele în timp real care nu pot fi accesate permanent, spre exemplu când comunicația cu un dispozitiv se întrerupe temporar. Tehnologia OPC clasică tratează aceste probleme furnizând mărci de timp și de calitate pentru datele transmise. Calitatea specifică dacă datele sunt precise (bune), nevalide (proaste) sau necunoscute.

Specificația OPC Alarm & Events - furnizează o interfață flexibilă pentru procesul de transmitere a alarmelor și a evenimentelor de la diferite surse de evenimente.

Pentru a primi notificări, clientul OPC A&E se conectează la server, subscrie notificările și apoi primește toate notificările declanșate în server. Pentru a limita numărul acestor notificări, clientul OPC poate specifica cu certitudine criteriile de filtrare.

Clientul OPC se conectează prin crearea unui obiect OPCEventServer în serverul OPC A&E, în prima etapă, și prin generarea unui obiect OPCEventSubscription folosit pentru a primi mesajele evenimentelor, în a doua parte. Filtrarea pentru aceste mesaje de evenimente poate fi configurată separat pentru fiecare subscriere. Figura 2 ilustrează diferitele obiecte create de clientul OPC în server.

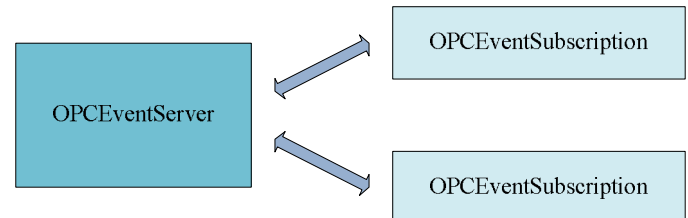


Fig. 2. Obiecte create de către clientul OPC pentru a primi evenimente

Specificația OPC Historical Data Access - Atâta timp cât OPC Data Access oferă acces în timp real, schimbând continuu datele, OPC Historical Data Access furnizează acces la datele stocate. De la simple detalii de autentificare serială în sistem și până la un sistem SCADA complex, arhivele istorice pot fi preluate în mod uniform.

Clientul OPC se conectează prin crearea unui obiect OPCHDAServer în serverul OPCHDA. Acest obiect oferă toate interfețele și metodele de citire și actualizare istorică a datelor. Un al doilea obiect, și anume OPCHDABrowser este definit pentru navigarea prin spațiul de adresă al serverului HDA.

Principala funcționalitate este citirea datelor istorice, realizată prin trei moduri diferite. Primul mecanism citește datele brute din arhivă, unde clientul a definit una sau mai multe variabile, iar domeniul timp este responsabil cu citirea datelor. Serverul returnează toate valorile arhivate pentru intervalul de timp specificat până la numărul maxim de valori definite de client. Al doilea mecanism citește valorile pentru

una sau mai multe variabile specificate de mărcile de timp. Al treilea mecanism de citire calculează valorile agregate datelor din baza de date a istoricelor pentru domeniul de timp specificat pentru una sau mai multe variabile. Valorile includ întotdeauna calitatea asociată și marca de timp. În plus, față de metodele de citire, OPC HDA, de asemenea, definește metode pentru inserarea, înlocuirea și ștergerea datelor din istoricele bazei de date.

Specificația OPC XML-Data Access - a fost primul caiet de sarcini OPC independent de platformă înlocuind tehnologia COM/DCOM cu HTTP/SOAP și servicii web.

Deoarece tipicele servicii web sunt independente, funcționalitatea a fost redusă la un set minim de metode pentru schimbul de informații OPC Data Access, fără de a fi nevoie de metode pentru a crea și modifica un context de comunicare. Erau necesare doar opt metode pentru a acoperi caracteristicile cheie ale OPC DA.

OPC XML-DA a fost proiectat pentru accesul la internet și integrarea în întreprinderi. Totuși, fiind independent de platformă, a fost pus în aplicare în principal în sisteme înglobate și pe platforme diferite de cele de la Microsoft. Din cauza consumului ridicat de resurse și performanțelor limitate, nu a avut parte de succesul așteptat pentru acest tip de aplicații. [1] [2] [3]

IV. MOTIVATIA PENTRU OPC UA

Primul și cel mai de succes standard clasic OPC, **OPC Data Access**, a fost proiectat ca interfață pentru comunicarea driverelor, permițând standardizarea citirii și scrierii accesului la datele curente ale dispozitivelor automatizate. Cazul de utilizare major a fost la sistemele HMI și SCADA, și anume, accesarea datelor provenite din diferite tipuri de hardware automatizate și dispozitive de la diferiți vânzători, care foloseau o interfață software definită de aceștia. Standardele care au urmat, **OPC Alarm & Event** și **OPC Historical Data Access**, au fost, de asemenea, proiectate pentru accesarea informațiilor provenite de la sistemele SCADA.

Odată cu succesul adopării a mii de produse de către OPC, acesta este folosit astăzi ca interfață standardizată între sistemele automatizate la diferite nivele ale piramidei de automatizare. Este folosit chiar în foarte multe domenii pentru care nu a fost proiectat, deși sunt încă multe alte domenii pentru care producătorii vor să utilizeze un standard ca OPC, însă limitările accesului la distanță nu le permite asta.

OPC XML-DA a fost prima abordare a Fundației OPC pentru a menține caracteristicile de succes ale OPC, dar de a utiliza, în același timp, un furnizor și o platformă de comunicare neutre de infrastructură. Există mai multe motive pentru care, tocmai crearea versiunii cu servicii web a specificației de succes OPC, nu acoperă cerințele pentru o nouă generație OPC. Un motiv a fost performanța slabă a XML Web Service în comparație cu versiunea originală bazată pe COM. În plus, folosind diferite stive XML Web Service intervin probleme de interoperabilitate.

Dar, pe lângă problema independenței de platformă, membrii companiei OPC au introdus cerința de expunere a

datelor și sistemelor complexe, eliminând restricțiile impuse de OPC-ul clasic.

OPC Unified Architecture s-a născut din dorința de a crea un înlocuitor adevărat pentru toate specificațiile existente bazate pe COM, fără pierderea unor caracteristici sau a performanței. În plus, acesta trebuie să acopere toate cerințele sistemelor independente de platformă interfațate cu capacități de modelare bogate și extensibile, fiind în măsură, de asemenea, să descrie sistemele complexe. O gamă largă de aplicații în care OPC-ul este utilizat, necesită scalabilitate pentru sistemele înglobate, și anume SCADA și DCS, până la sisteme MES și ERP[1] [2] [3].

V. CERINȚE IMPUSE SPECIFICAȚIEI OPC UA

Cerințele pot fi grupate în cele pentru comunicarea dintre sistemele distribuite care sunt capabile să schimbe informații și cele pentru modelarea datelor care descriu un sistem și informațiile disponibile.

TABELUL 1 CERINȚE PENTRU OPC UNIFIED ARCHITECTURE[1] [2]

Comunicația între sistemele distribuite
Fiabilitate Robustețe și toleranță la greșeli Redundanță Independență de platformă Scalabilitate Performanță înaltă Internet și firewall-uri Securitate și controlul accesului Interoperabilitate
Modelarea Datelor
Model comun pentru toate datele OPC Orientat-pe-obiect Extensibilitatea tipurilor din sistem Meta informații Date complexe și metode Scalabilitate de la modele simple la modele complexe Model de bază abstract Bază pentru alte modele de date standard

VI. CONCEPTELE DE BAZA ALE MODELULUI OPC UA

Conceptele de bază ale modelului OPC UA sunt *Nodurile* (*Nodes*) și *Referințele dintre ele* (*References*). Nodurile pot aparține diferitelor Clase de Noduri (*NodeClasses*), depinzând de scopul acestora.

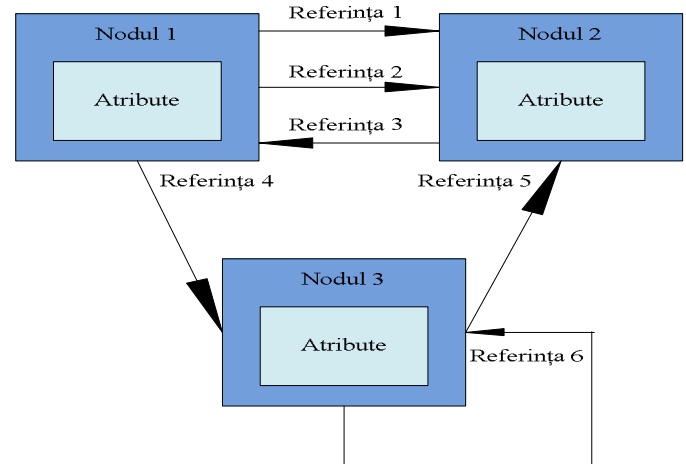


Fig. 3. Nodurile și Referințele dintre Noduri

Aceste concepte reprezintă: fie instanțe, fie diferite tipuri de date. *Atributele* sunt folosite pentru a descrie *Nodurile*, iar în funcție de *NodeClass*, un *Nod* poate avea diferite seturi de *Atribute*. Figura 3 ilustrează aceste concepte de bază. *Nodurile* 1, 2 și 3 conțin câte un set de *Atribute*, iar legăturile dintre aceste noduri sunt realizate de *Referințe* (1..6).

Există *Atribute* comune mai multor *Noduri*. Aceste *Atribute* sunt detaliate în tabelul următor.

TABELUL 2 ATRIBUTELE COMUNE NODURILOR DIN OPC UA

Atribute	Descrierea
NodeId	Reprezintă identificatorul unic al Nodului în interiorul serverului OPC UA și este folosit pentru a adresa un nod în cadrul Serviciilor OPC UA. Acesta nu este localizat.
NodeClass	Reprezintă o enumerare de identificare a Clasei de Noduri a Nodului curent (Obiect sau Metodă).
BrowseName	Identifică Nodul care navighează prin serverul OPC UA. Acesta nu este localizat.
DisplayName	Conține numele Nodului care ar trebui utilizat pentru a afișa numele într-o interfață de utilizator. Prin urmare, este localizat.
Description	Acest Atribut opțional conține o descriere textuală a localizării unui Nod.
WriteMask	Este opțional și specifică care dintre Atributele unui Nod pot fi scrise (writable) sau modificate (modified) de un client OPC UA.
UserWriteMask	Este opțional și specifică care dintre Atributele unui Nod pot fi modificate de un utilizator conectat la server.

NodeId identifică în mod unic un *Nod* în server și este cel mai important concept de adresare a informației. Serverul returnează Id-urile Nodurilor (*NodeIds*) care accesează sau interoghează Spațiul de Adrese, iar clientul folosește *NodeId*-ul pentru a adresa *Nodul* în apelurile Serviciilor. Un *nod* poate avea mai multe *NodeId*-uri alternative folosite pentru adresare. Identificatorul nodului (*NodeId* - ul) standard poate fi dobândit prin citirea atributului *NodeId*, chiar dacă *Nodul* a fost accesat de un *NodeId* alternativ.

BrowseName este folosit doar pentru navigarea intenționată și nu pentru a afișa numele unui *nod*.

DisplayName și *Description* sunt utilizate în scopul localizării.

Teoretic, o *referință* reprezintă relația dintre exact două noduri. Prin urmare, aceasta este identificată în mod unic de: sursa nodului referință, nodul țintă, tipurile semantice de referință și direcția referinței.

În practică, serverul poate expune o *referință* doar într-o direcție, iar *referința* poate indica un *nod* din cadrul altui server OPC UA sau un *nod* inexistent. În concluzie, este corect să credem că, o referință este la fel ca un pointer care indică un alt nod, acesta din urmă reținând *NodeId*-ul pentru un altul. Acest lucru se poate urmări și în figura următoare.

Se disting două tipuri de *referințe*, și anume: referințe simetrice și asimetrice. O referință asimetrică este reprezentată printr-o săgeată care arată o direcție, spre exemplu o referință de tipul "has-parent" sau "is-child-of" care indică nodul părinte sau copil. Referința simetrică, în schimb, este reprezentată printr-o săgeată dublă indicând ambele direcții. Un astfel de exemplu este "is-sibling-of". În cazul referințelor

asimetrice, direcția este foarte importantă, pe când în cazul celor simetrice, aceasta nu contează.

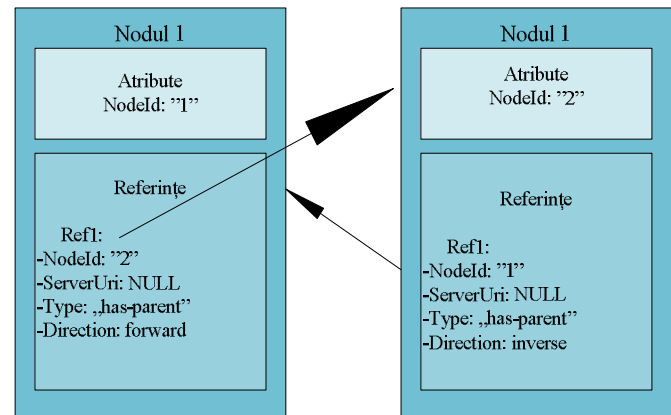


Fig. 4. Referințele, văzute ca pointeri la Noduri

O referință nu poate fi accesată în mod direct, ci doar indirect prin intermediul unui nod. De asemenea, aceasta nu poate conține atribute sau proprietăți și nu poate fi reprezentată de un nod. În concluzie, referințele sunt utilizate pentru a expune diferențele semantice ale modului în care sunt accesate nodurile. În acest sens, specificația OPC UA a definit un set de *tipuri de referințe* (ReferenceTypes). Câteva dintre acestea, sunt folosite în locurile fundamentale, spre exemplu, la descrierea unui tip de ierarhie. De aceea, acest concept este considerat, ca fiind unul extensibil, deoarece permite definirea propriilor tipuri de referințe cu un specific anume. Pentru a organiza aceste tipuri de referințe, ele au fost gestionate într-o ierarhie.

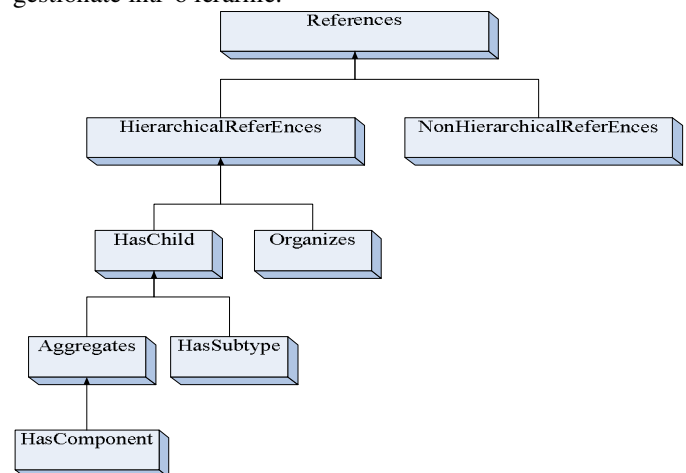


Fig. 5. Ierarhia tipurilor referință de bază

VII. STUDIU DE CAZ

Aplicația dezvoltată pentru acest studiu de caz, a avut la bază realizarea unei aplicații practice între calculator și un sistem înglobat (embedded) - *eBox-2300SX*. Aplicația constă în portarea clientului OPC pe sistemul înglobat *eBox-2300SX* folosind platformele *.NET Compact Framework 2.0/3.5* și *Windows Embedded CE 6.0*.

Această aplicație a fost dezvoltată pentru a permite

monitorizarea și controlul proceselor industriale. Din acest motiv, aplicația a fost denumită *MCPI* (Monitorizarea și Controlul Proceselor Industriale).

Aplicația *MCPI* este un modul software executabil, care face parte dintr-o arhitectură mai complexă, a unei aplicații *SCADA*. Aplicația *MCPI* prezintă două moduri de lucru. Primul este modul de lucru editare, care permite implementarea și modificarea proiectului. Acest mod de lucru se realizează atunci când aplicația este decuplată de la servere. Al doilea mod de lucru este cel în execuție. În acest caz, aplicația execută ceea ce a fost proiectat.

Componentele arhitecturii aplicației *MCPI* sunt: *proiectul* (reprezentat de modulul software executabil), *obiectele* și *conexiunile*. Sarcina utilizatorului este de a crea, configura și conecta obiectele între ele.

Proiectul este un grup de obiecte independente sau interconectate între ele. Utilizatorul creează proiecte (aplicații *MCPI* distincte) pentru a îndeplini anumite sarcini. Utilizatorul poate deschide și închide proiecte fără a influența celelalte proiecte care rulează în paralel.

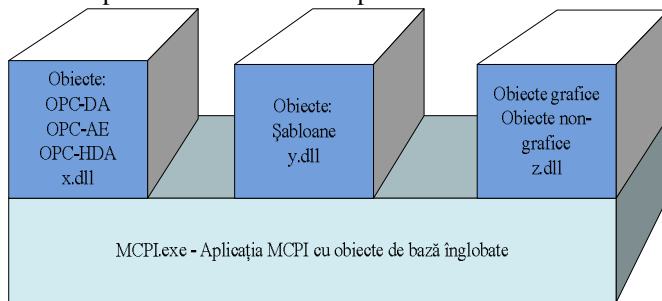


Fig. 6. Arhitectura aplicației *MCPI*

Obiectele de tip OPC sunt singurele obiecte care permit conectarea la serverele OPC. Deoarece arhitectura aplicației *MCPI* este bazată pe obiecte și conexiuni între obiecte, odată creat un obiect OPC, la el se vor putea conecta oricare alte obiecte din cadrul unui proiect. Aceste obiecte pot fi afișoare grafice, obiecte de control, prin intermediul cărora operatorul uman are acces la valorile din proces.

Fiecare obiect are interfața "*BaseObject*" și opțional poate să aibă *interfața OPC* sau o *interfață specifică* așa cum s-a menționat anterior.

Totalitatea obiectelor unui proiect formează *spațiul de adrese al unui proiect*. Există aici o excepție și anume obiectele grafice, de afișare a unui text sau a unei imagini nu sunt incluse în spațiul de adrese al unui proiect fiind gestionate doar ca elemente de afișare a informațiilor la nivel de fereastră.

Un proiect va fi un ansamblu de obiecte și conexiuni, astfel, proiectul va trebui să gestioneze o listă de obiecte interconectate sau independente. Obiectele vor fi grupate în directoare. Fiecare obiect are un nume unic în cadrul directorului, iar directoarele vor avea un nume unic la nivel de proiect. Deci proiectul va fi o structură arborescentă de directoare, subdirectoare și obiecte. Prin urmare, vom considera un director principal **director rădăcină** al proiectului, și în acesta se vor crea obiecte, directoare, subdirectoare, etc.

Directorul principal nu va apare în structura arborescentă a spațiului de adrese al proiectului. El va fi controlat intern pentru a asigura o structură uniformă a modului de gestionare a obiectelor proiectului [5] [6] [7] [8].

VIII. CONCLUZII

Modelul spațiului de adrese reprezintă o îmbunătățire remarcabilă a OPC UA, față de vechile specificații.

Transportul definește diferite mecanisme optimizate pentru diferite cazuri de utilizare.

Modelarea datelor definește regulile și blocurile de bază necesare pentru a expune un model informațional cu OPC UA.

Aplicația dezvoltată pentru studiul de caz, a avut la bază realizarea unei aplicații practice între calculator și un sistem înglobat *eBox-2300SX*. Aplicația constă în portarea clientului OPC pe sistemul înglobat *eBox-2300SX*.

Această aplicație a fost dezvoltată pentru a permite **monitorizarea și controlul proceselor industriale - MCPI**.

Aplicația este în totalitate orientată pe obiecte. Obiectele din cadrul aplicației sunt reprezentate de un modul software cu funcționalitate internă, care are un set de mărimi de intrare, memorie internă și un set de mărimi de ieșire, rezultate ca urmare a prelucrărilor interne [5] [6] [7] [8].

ACKNOWLEDGMENT

Această lucrare a beneficiat de suport financiar prin proiectul "**Provocările cunoașterii și dezvoltare prin cercetare doctorală PRO-DOCT Contract nr. POSDRU/88/1.5/S/52946**", proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013.

BIBLIOGRAFIE

- [1] Frank Iwanitz, Jurgen Lange. OPC Fundamentals, Implementation and Application, . Heidelberg - Huthig: 2nd rev. Edition, ISBN 3-7785-2883-1, 2002.
- [2] OPC-Foundation. OPC UA Specification: Part 3 – Address Space Model. Version 1.00. July 28, 2006.
- [3] OPC-Foundation. OPC UA Specification: (Part 1-11) – Concepts. Version 1.00,. July 28, 2006.
- [4] (IANA), Internet Assigned Numbers Authority. IPv4 Address Space Registry. 2009, March 11.
- [5] Gherasim Simona-Anda, Mihai-Florentin Ursuleanu, Ovidiu Gherman. A Programming Model for Context - Sensitive Application. Iași, România: 10th RoEduNet IEEE International Conference Proceedings - Networking in Education and Research , 2011.
- [6] Gherasim Simona-Anda, Găitan A.M., Găitan Gheorghiță Vasile, POPA V. New trend on OPC Middleware. Universitatea „Ștefan cel Mare” Suceava, România: International Conference on Development And Application System, Mai 2010.
- [7] Gherasim Simona-Anda, Găitan Gheorghiță Vasile,. An overview about state of art for Middleware. Universitatea „Ștefan cel Mare” Suceava, România: Seminarului Științific național „Sisteme Distribuite”, decembrie 2009.
- [8] Găitan Gheorghiță Vasile, Dănilă M. G., Robu M. G., Găitan N. C. MCPI – a HMI Application for Monitoring and Controlling Industrial Processes. . Suceava-Romania: The 9th International Conference on Development and Application Systems, ISSN 1844-5020, pp.32-37., 2008.