

# Industri Automatizazioan kontrol-programak berrerabiltzeko bidean

*Edurne Irisarri, Elisabet Estévez eta Marga Marcos*

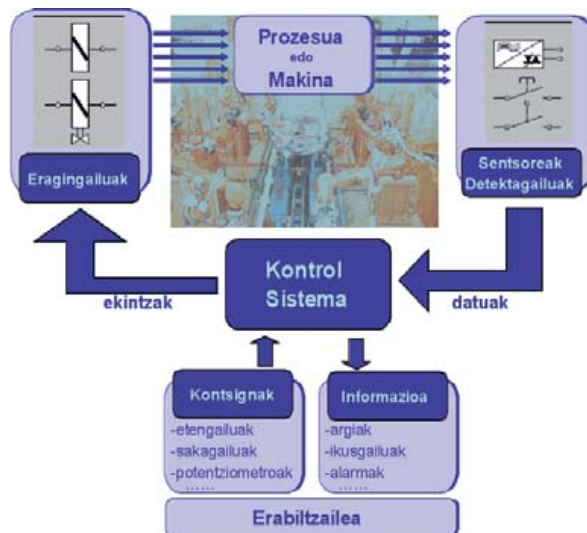
Sistemen Ingeniaritza eta Automatikako Saila,  
Euskal Herriko Unibertsitatea UPV/EHU  
edurne.irisarri@ehu.es, elisabet.estevez@ehu.es, marga.marcos@ehu.es

**Laburpena:** Gaur egungo industri aplikazioek sistema konplexuagoak, seguruagoak eta fidagarriagoak behar dituzte, bai eta malgutasun eta berrerabilgarritasun handiagokoak ere. Hori dela-eta, lehenasuna hartu dute garapen-ingurune ezberdinen integrazioak eta alde zuzeneko beste aplikazioetarako sortutako kontrol-programen kodearen berrerabilgarritasunak. Fabrikatzaileek sistemak programatzeko sistemetan IEC 61131-3 estandarrari jarraitzean datza helburu hauek lortzeko giltzarria. Honela, automatizazio-proiektuen egitura eta programatzeko lengoaiak batuz, kodea berriz erabiltzea lor daiteke. Benetako berrerabilerak, estandarrak definitzen dituen alderdiez gain, ingurune batetik beste batera transferitzeko mekanismoak eskatzen ditu. Lan honetaz diharduen PLCopen erakundeak XML interfaze bat definitu du. Interfaze hau daukaten programazio-inguruneetan IEC 61131-3 jarraituz gero, zuzenean lortzen da elkarreragina eta, ondorioz, kodea berrerabiltzea ere. Lan honetan gehiago eskaintzen duen bidea aurkezten da, edozein programazio-inguruneraren elkarreragin lortzeko metodologia aurkezten baita, erabiltzaileak honela kontrol programen berrerabilera benetan aukeran izan dezan.

**Abstract:** Current industrial applications demand the design of more and more complex, safe and trustworthy control systems that exhibit a high degree of flexibility and reusability. Owing to this, the integration of different development environments and the reusability of control logic code developed previously for other applications become priority objectives. The key for these goals lies in the compliance with the IEC 61131-3 standard by the controller programming systems developed by the different manufacturers. In fact, making use of the automation project model and programming languages defined by the standard the code reusability can be achieved. However, as well as these standard issues and in order to achieve a true reuse of application code, procedures for transferring the information programmed in one specific programming environment to others are required. In this sense, PLCopen organization has defined an open XML interface to achieve interoperability between different development environments. In case of programming environments that are both provided with this interface and compliant with IEC 61131-3, the interoperability and thus, code reusability, are directly achieved. This work goes a step further as it presents a methodology that makes possible the interoperability among any programming environment. This could be very useful as it affords the final users the necessary resources to achieve the code reuse in any case.

## 1. SARRERA

Gaur egun, industri arlo gehienetan prozesuek eta makinek eskatzen duten konplexutasuna, erantzun azkarra eta osagaien banaketa direla-eta, portaera kontrolatzeko lanetan nekez aurkitzen da automatizazio-teknikak eta ekipoak integratzen ez dituen plantarik. Izan ere, plantak automatizatzean (ikus 1. irudia), lehenik, sentsore eta detektagailu egokiak jarritz ahalbidetzen da ekoizpeneko edo beste edozein motatako aldagaien neurketa automatikoa, hots, gizakiak esku gabekoa. Bigarrenik, erabiltzaileak prozesurako edo makinarako emandako argibideak jarraituz eta programa batek jasotzen dituen kontrol-aginduak kontuan hartuz, plantatik jasotako datuak prozesatu egiten dira. Azkenik, sistemak berak plantaren portarari eragiteko erabakiak hartu eta gehitutako eragingailuetan dagozkien ekintzak sortarazi; aldi berean, erabiltzaileari ere portaeraren garapenaren berri emateko aukera eskaintzen zaio. Beraz, bistan daude automatizazioak ekartzen dituen abantailak, prozesuak bizkortzea, ekoizpena ugaltzea, langilearen lana arintzea eta eskuz ezin burutuzko lanak garatzea. Automatizazio lan hauetan, produkzio eta prozesuaren hobekuntzaz gain denborak eta kostuen murrizketa ere lortzen dira, elektronika, informatika eta mikroprozesagailuen teknologietako lorpenak jasotzen dituzten Industri Prozesuak Neurtzeko eta Kontrolatzeko Sistemak (IPNKS) erabiltzearen ondorioz.



**1. irudia.** Prozesu edo makina baten automatizazioa

Azken hamarkadetan, ordea, sistema hauen erabileraren hazkundera [1] oso handia eta azkarra izan da industri aplikazioetan eta horren ondorioz, hobekuntza bakoitzerako automatizazio-produktu hauen fabrikatzaile bakoitzak irtenbide bana proposatu izan du eta aniztasuna asko areagotu da. Egoera hau

ez zaie enpresei batere onuragarria gertatu; izan ere, ekoizpen-lerroan aldaketa edo eguneraketa ezarri behar den bakoitzean diru-inbertsioak egin behar izaten dira.

Aldiz, kontsumitzaileak gero eta produktu hobe eta merkeagoak eskatzeaz gain, merkaturaren bereganatzeko sortzen den lehiaketa izugarriak ekoizpen-lerroen etengabeko eguneraketarako politika sortarazten du enpresetan. Helburu hauek lortzeko, hots, gero eta aldakorragoa eta lehiakorragoa den merkatura egokitzeko, ezinbestekoa da kontrol-sistematan ekipoen integrazioa, aplikazioen berrerabilera, malgutasuna eta hobekuntza lortzea.

Enpresa baten arrakasta garai berrietara azkar egokitzeko duen gaitasunaren arabera gertatuko da. Hau lortzeko, ezinbesteko laguntza ematen dute fabrikatzaile ezberdinen produktuen elkarreragina ahalbidetzen duten irtenbide irekietan eta estandarren erabilerak. Automatizazioaren arloan irtenbide irekietako katean maila txikia bada ere, IEC 61131-3 [2] izeneko estandarren aplikazioa garrantzi handikoa gertatzen ari da, edozein enpresetako sistema automatikoetan berehalako onurak ematen baititu benetan. Arau hau da kontrol-aplikazioak programatzeko gehien erabiltzen den estandarra. Estandar honek industri prozesuen neurri eta kontrol sistemak programatzeko software-eredua eta bost lengoaiaren gramatika zehazten ditu. Normaldutako lengoia hauek eta automatizazio-proiektua antolatzekeko ereduak, besteak beste, kode berrerabilgarria garatzea dute helburu, hots, aplikazioen kontrol-programen berrerabilera.

Aplikazio-kodearen benetako berrerabilgarritasuna lortzeko ezinbestekoa da, ordea, sistema jakin batean programatu den informazioa beste sistemetara transferitzeko gaitasuna, ekipo ezberdinen software-elkarreragina alegia. Halaber, beharrezkoa da, informazioak, testuzkoa zein grafikoa, sortu den formatu berean jarratzen duela bermatzea. Ildo honetan, aipatzekoa da produktutik eta fabrikatzailearengandik beregaina den PLCopen [3] nazioarteko erakundearen lana. Honekin, kontrol programazioarekin lotutako gaiak ebazteko irizpide nagusi gisa, estandarren erabilera bultzatu nahi da. Jorrotutako lanen artean, besteak beste, elkarreragina lortzeko interfaze ireki bat proposatu da. Interfaze honek ahalbidetzen du kontrol-sistema programatzeko tresnaren pantailan aurkeztutako IEC 61131-3 elementuak ezberdinak diren beste ingurune batzuetara XML (*eXtensible Markup Language*) teknologiaz baliatuz transferitzea. Modu honetan, automatizazio aplikazioetan fabrikatzaile ezberdinen sistemak elkarrekintzan jartzeko aukeraren atea irekitzen zaie enpresei.

Lan honetan lehenik eta behin, prozesu automatizatuetan erabiltzen diren kontrolagailu logiko programagarriak eta jasan duten garapena

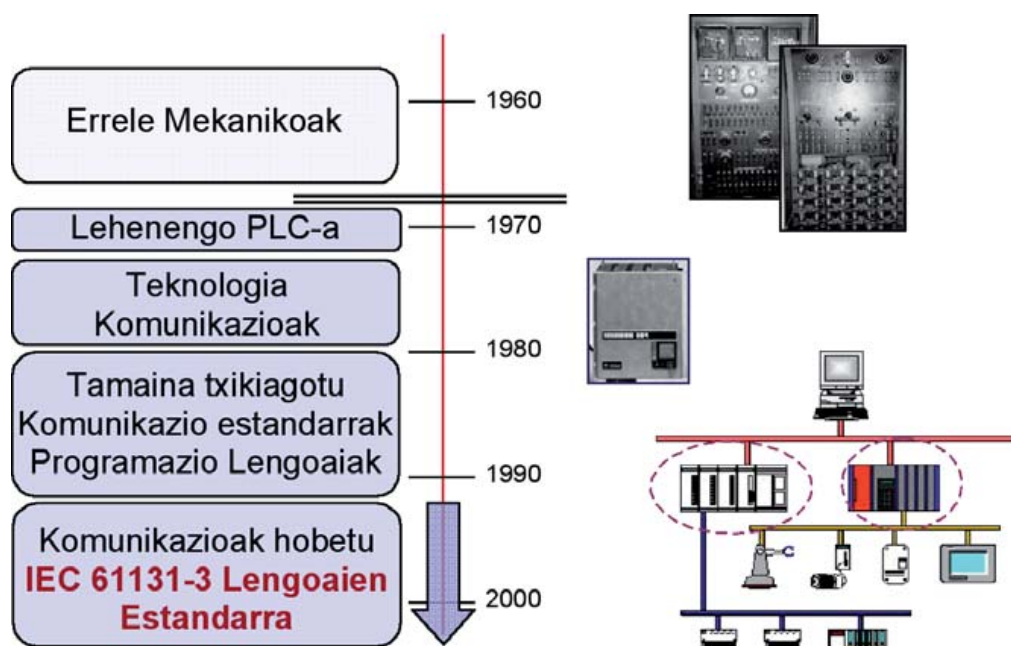
azaltzen da; bigarrenik, garapen honek ekarri duen estandarizazio beharraz eta IEC 61131-3 arauaren ezaugarri nagusiez mintzatuko gara; bukatzeko, estandarra abiapuntu hartuta kontrol-programak elkartrukatzeko PLCopen-ek definitu duen interfazearen gaia jorratuko dugu.

Ikuspuntu honetatik, kanpo aplikazio baten diseinua azaltzen da; aplikazioak aukera ematen du garapen-ingurune batek informazioa transferitzeko XML teknologien bidez definitutako interfaze hau erabiltzeko. Hau laguntza handikoa izango da kode-garatzailleentzat, PLCopen-en interfazea onartzen ez duten programazio-tresnen kasuetan ere kodearen berrerabilgarritasuna lortzeko.

## **2. KONTROLAGAILU LOGIKO PROGRAMAGARRIEN GARAPENA**

Industria arloan azkeneko hamarkadetan gertatu izan den garapena ezinezkoa gertatuko zen kontrolagailu logiko programagarriak (*PLC: Programmable Logic Controller*), edota industriako automata programagarriak ere deitutako ekipo elektronikoak, izan ez bagenu. Izan ere, sistema elektroniko hauek industriako ordenagailu espezializatuak direla esan genezake [4][5]. PLC-ak industri prozesuak, makinak edota plantak programatzeko eta kontrolatzeko erabiltzen dira. Helburu hau lortzeko ekipoak prozesutik edo makinatik informazioa jasotzen du sentso-re, dektore edota etengailuen bidez. Ekipoaren baitan grabatzen den logika edo programari esker seinale egokiak sortzen ditu eta honela kanpoko prozesuari edo makinari eragiten dio.

PLC-ak, baldintza gogorretan, tenperatura altu edo baxuetan edota zarata elektriko handiko industri inguruneetan aritzeko kontrol-sistema egokituak dira. Kontrol-teknologia hau, gainera, sarrera eta irteerako interfaze anitz izateko eta sentso-reak eta eragingailuak diren bezalako osagai mekatronikoak konektatzeko eta kontrolatzeko diseinatuta dago. Era berean, aipatzekoa da irtenbidea aplikazioaren programa-baldintza berrietarako aldatu eta egokitzea nahikoa izan daitekeela makinaren edo prozesuaren portaeran aldaketak edo eguneraketak egiteko. Modu honetan, PLC-a sortu aurretik aplikazio hauetarako erabiltzen ziren logika kableatuko errele mekanikoetako ekipo elektrikoekin alderatuz (ikus 2. irudia), denbora eta kostuak nabarmen murriztu ez ezik, ahalbidetzen da makina eta prozesu bat baino gehiago aldi berean kontrolatzea ere.



## 2. irudia. PLC-en garapena historian zehar

PLC-en ustezko asmatzailea Dick Morley ingeniaria da. Berak, 60ko hamarkadan, ikusi zuen errele-koadro handien logika kableatuaren garaia gainditu beharra zegoela eta era abantalaitsua izango zela bere ordeztu programagarria izan behar zuen osagai txikiagoa erabiltzea. Lan hau ez zen izan batere erraza; izan ere, industri inguruneen baldintza gogorretan sendotasuna, fidagarritasuna eta malgutasuna eskaintzeaz gain, enpresa erosleetako langileek programa lezaketan produktua bilatzen zen. Hurrengo hamarkadan, eremu-busez baliatuz eta PLC-en artean komunikatzeko lortu zen hobekuntzari esker, ahalbidetu zen kontrolagailua kontrolatu beharreko sistematik urruntzea eta honela kontrola banatzea. PLC-en komunikazio sistemak hobetzearekin batera, garatzen joan dira gero eta konplexuagoak bilakatu diren industri aplikazioak kontrolatzeko programak gorde ahal izateko memoria handiagoak eta bai aukera gehiago eskaintzen dituzten PLC-ak programatzeko lengoaiak ere. 80ko hamarkadako hobekuntzak nabarmen bideratu dira PLC-en tamaina txikiagotzera, prozesatzeko abiadura azkarragoak lortzera, kontrol-teknika konplexuagoak aplikatzera eta lengoaiak berriak sortzera.

Hain izan zen zabala eta azkarra PLC-en inguruko hobekuntzak ezen fabrikatzaile bakoitza garatzen joan zen komunikazio-protokolo zein programatzeko lengoaiak propioak. Gauzak honela, kontrol-ingeniariak zailtasun handiei aurre egiteko beharrez aurkitu ziren. Esaterako, fabrikatzaile ezberdinetako PLC-en arteko elkarrekintza, alde aurretik aplikazio

baterako sortutako programaren kodea berrerabiltzea, edota programatzeko modua erraztea ziren kontrol-ingeniarien lana arintzeko gainditu beharreko oztopoak.

Aurreko horrek guztiak behartu zituen teknikariak PLC-en inguruko hardwarerako zein softwarerako estandar bat sortu eta konfigurazio eta programazio-lengoaiei dagozkien xehetasunak zehaztuta uztera. 1993ko urtean, Nazioarteko Batzorde Elektroteknikoak (IEC: *International Electrotechnical Commission*) argitara eman zuen IEC 61131 delako Kontrolagailu Programagarrietarako Estandar Nazioartekoa ize-neko arautegia.

### 3. IEC 61131-3 ESTANDARRA

Sistemen elkarreragina da automatizazio arloan estandarizatzeko arrazoi nagusietako bat. Honekin batera, programatzeko modua errazteko eta aldez aurretik sortutako kodea berrerabiltzeko beharrek IEC 61131 estandarra sortzera eraman zuten. Hau izan zen automata programagarien eta gehitzen zaizkien periferikoen estandarizaziorako lehenengo urratsa.

IEC 61131 araua hurrengo zortzi zatietan dago banatuta [4][5]:

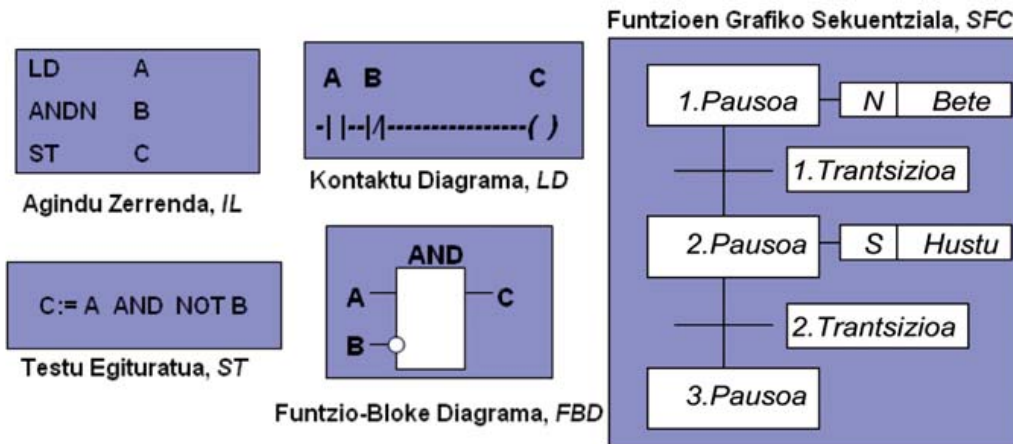
- **1. zatia.** Informazio orokorra: estandarreko oinarritzko terminologia eta kontzeptuen definizioak ematen ditu.
- **2. zatia.** Hardwarea: Ekipoen eta proben betekizunak. Eraiketa mekanikoa eta elektrikoa eta egiaztatze probak.
- **3. zatia.** Programazio-lengoaiak: Software-egitura, lengoaiak eta programen exekuzioa PLC-an.
- **4. zatia.** Erabiltzailearentzako gida-lerroak: PLC-ak aukeratzeko, instalatzeko eta mantentzeko gidak.
- **5. zatia.** Komunikazio-zerbitzuak. Beste gailuekiko elkarreragiketan laguntzeko komunikazio-zerbitzuak, manufaktura-sistemetako mezu-zerbitzuetan oinarrituak.
- **6. zatia.** Ereku-bus bidezko komunikazioak: Komunikazio-zerbitzuak IEC eremu-busak erabiliz.
- **7. zatia.** Kontrol lausoaren programazioa: PLC-az baliatuz logika lausoa erabiltzeko softwarea, funtzio-bloke estandarrak ere barne duena.
- **8. zatia.** Kontrol programagarrietarako lengoaiak implementatzeko laguntzak: IEC 61131-3 zatiko lengoaietarako laguntzen aplikazioa eta implementazioa.

IEC 61131 arauaren hirugarren zatiak [2] IPNKS-etan aplikatzeko software-eredua eta programatzeko lengoaiak jorratzen ditu. Honek programazio-teknikak PLC-aren fabrikatzailearengandik beregaina bihurtzen dituzenez, abantaila ugari eskaintzen dizkie, zalantzarik gabe, kontrol-sistemen ingeniariari. Izan ere, industri automatizazioan programatzeko lengoaien estandarizaziorako benetako oinarria izan da hau.

Estandarraren zati honek PLC-ak programatzeko bost lengoaien gramatika, sintaxia eta semantika, zehazten ditu. Zehazki, 3. irudiko adibideetan ikus daitekeen bezala, lengoia horietako bi testuzkoak dira, bi grafikoak, eta azkenekoa testuzkoa zein grafikoa izan daitekeen automatizazio-aplikazioa egituratzeko lengoia da. Lengoaiak ondoko hauek dira:

- Kontaktua Diagrama (LD, *Ladder Diagram*) da historikoki PLC-en hasieran erabilia izan zen lengoia, sareetan antolatutako sinbolo grafikoetan oinarritzen baitzen; hala egiten zen ekipo kableatuetako erreleen logikako diagrametan, hain zuzen ere. Seinale Boolearretako diseinuetan aplikatzeko egokia da batik bat.
- Aginduen Zerrenda (IL, *Instruction List*) maila baxuko lengoia testuzkoa da, lengoia mihizatzailearekin antza handikoa alegia. Testu lerroetan oinarritzen da, lerro bakoitzean eragiketa bat egiteko agindua jasotzen delarik.
- Funtzio-Bloke Diagrama (FBD, *Function Block Diagram*), funtzioak, funtzio-blokeak edo programak aurkezten dituzten objektu grafikoen edo blokeen bidez prozedura konplexuak programatzeko erabiltzen da, zirkuitu elektronikoen diagrametan egiten den antzera. Prozesuetan oinarritutako industrian oso erabilia da.
- Testu Egituratua (ST, *Structured Text*), lengoia aurreratu eta berriena da, goi mailako lengoia, eta blokeetan egituratua. Pascal lengoia gertu dago eta ADA eta C lengoaien eragina jaso du. Prozesu konplexuak automatizatzeko oso erabilia da.
- Funtzioen Grafiko Sekuentzialak (SFC, *Sequential Function Chart*), programa eta funtzio-blokeen bidez automatizazio aplikazio baten egituraketa sekuentziala osatzea du helburu. Grafikoki edota testu bidez programa daiteke.

Programatzeko lengoaietz gain, IEC 61131-3 zatiak automatizazio-proiektuaren egitura orokorra antolatzeko eredua definitzen du, software-eredua hain zuzen. Estandarrak definitutako ereduak automatizazio-proiektuaren kodearen modularitasuna du helburu, modu honetan alde aurretik inplementatutako eta probatutako kode-unitateen berrerabilera ahalbidetuz.



**3. irudia.** IEC 61131-3 estandarrak zehaztutako PLC-ak programatzeo lengoaiak

Estandarrak definitzen duen software-eredua elementu jakin batzuek osatzen dute. [6] Laburki, IEC 61131-3 arauak kontrol aplikazioa definitzeko zehazten dituen elementuak ondoko hauek dira: konfigurazio (*configuration*) izeneko elementuak PLC bakoitza aurkezten du. Konfigurazioak, aldiz, baliabideak (*resources*) eta atazak (*tasks*) dauzka. Baliabideak programaren exekuzioa egiten du eta atazek programaren zati ezberdinen exekuzioa antolatzeo aukera eskaintzen diote programatzaileari. Azkenik, baliabideek kodea exekutatzen duen programa antolatzeo unitate (POU, *Program Organization Unit*) izeneko elementuak ditu. POU motako elementua den programako kodea estandarrek zehaztutako bost lengoaietako edozeinetan idatzita egon daiteke eta funtzio-blokeak (*function blocks*, hauek ere POU motakoak) eta funtzioak (*functions*) erabiltzen ditu. Gainera, POU-ek definitu edo/eta erabiltzen dituzten aldagaiak (*variables*), mota, ikusgarritasun eta balio ezberdinetakoak izan daitezke. Orokorrean, baliabideetan exekutatzen diren programek beste POU motak ere erabil ditzakete aldagaien bidez.

POU-ak aplikazioaren kodea berrerabiltzeo funtsezko elementuak dira; izan ere, proiektuaren beste zatietatik beregainki konpilatu daitezkeen kode-unitateen moduluak dira. POU-en beregaintasunak ahalbidetzen du automatizazio lanetan gero eta modulu gehiago sartzea eta alde zuzenetik sortutako software-unitateak berrerabiltzea.

Estandarrak mekanismoak eskaintzen ditu software-ereduaren osagaiak eta automatizazio-aplikazioa definitzeko, baina ez dio ordea erabiltzaileari inportatzeo/esportatzeo formaturik zehazten. Hori dela-eta, aplikazio baterako sortutako kodea edo kode-unitateak beste programazio-ingurune



batzuetan benetan berrerabil dadin eskuz editatu behar da berriz. Hortaz, ezinbestekoa gertatzen da kodearen benetako berrerabilgarritasuna lortzeko PLC-ak programatzeko tresnen arteko elkarreragina, eta horixe izan da, hain zuzen ere, PLCopen erakundeak izan dituen helburuetariko bat.

#### **4. PLCopen ERAKUNDEA**

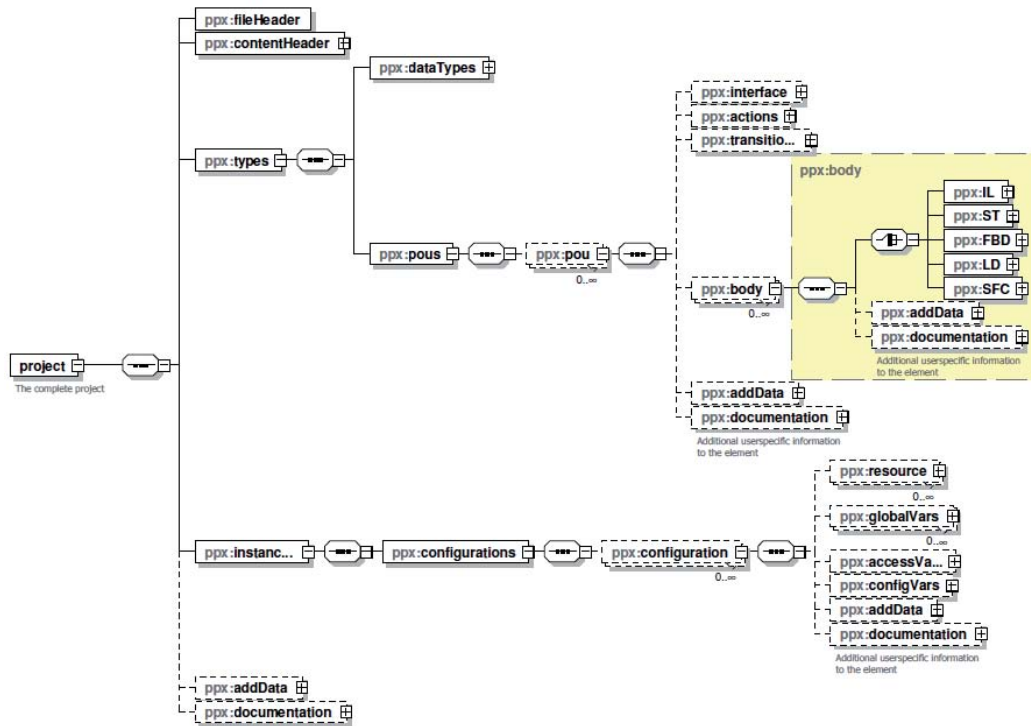
PLCopen [3] fabrikatzaileengandik eta produktuetatik beregaina den nazioarteko erakundea da. Kontrol-programazioaren inguruan sortutako arazoei irtenbideak ematea du helburu. Bere zeregin guztien artean, azpimarratzekoa da automatizazioaren arloan nazioarteko estandarrak gero eta erabiliagoak izan daitezen egiten ari den lana. Erakunde hau batzorde jakin batzuetan antolatuta dago; batzorde batzuk promozio lanetarako eta beste batzuk kontu teknikoetarako zuzenduta.

PLCopen erakundeko batzorde teknikoen artean, dagokigun gaia dela-eta, aipatzekoa da 6.Komite Teknikoak, XML-rako TC6 delakoa, egiten ari den lana. Izan ere, TC6-ak elkarrekintzan aritzeko interfaze ireki bat [7] definitu baitu. Interfaze honen definizioa markatzeko lengoia zabalgarriaz (XML, *eXtensible Markup Language*) [8] baliatuz gauzatu dute. Lan honi esker, PLC-a programatzeko erabiltzen den tresnaren pantailan dagoen informazioa beste ingurune batzuetara transferitzeko ahalmena eskuratzen da.

PLCopen-ek sortutako XML interfazeak aplikazioaren kodea programazio-ingurune batetik beste ingurune batera transferitzea du helburu. Informazioa galdu gabe transferitzeko kodea automatizazio-proiektu osoa, POU-ak, zein programatzaileak diseinatutako Funtzio Blokeetako liburutegiak izan daitezke.

Interfaze honek IEC 61131-3 estandarreko elementuak XML moduan definitzen ditu *XML schema* (.xsd) [9][10] delako teknologia erabiliz. Teknologia hau lengoia berri bat definitzen duten murrizte lexiko eta sintaktikoak jasotzen ditu. Hau da, lengoaiaren elementuak, ezaugarriak, motak, eta estilo arkitektonikoa ere, definitzen ditu. Aldi berean, teknologia honek ahalbidetzen du XML dokumentu bat ontzat hartzeko eragiketa egitea; izan ere, ondo osatuta egoteaz gain berari lotutako egitura eta murrizteak betetzen dituen egiaztatzeko erabiltzen da. Bereziki, lengoaiaren elementuak XML-n definitzeko, *World Wide Web Consortium* (W3C) [8] delakoaren *schemak* aukeran dituen *simpleType* eta *complexType* mekanismoak erabiltzen dira. Elementu hauen ezaugarriak, aldiz, murrizte eta aurkezpen ereduak *attribute*, *restriction* eta *representation pattern* direlako mekanismoez baliatuz definitzen dira. Estilo arkitektonikoa definitzeko WC3-ko *schemak* eskaintzen dituen beste hainbat elementu erabili dira, *sequence* eta *choice* direlakoak, alegia.

4. irudian aurkeztu bezala, aurreko teknologia da hain zuzen IEC 61131-3 estandarreko elementuak dituen XML *schema* definitzeko PLCOpen erakundeak TC6 komiteak erabiltzen duena<sup>1</sup>.



#### 4. irudia. PLCOpen-eko XML Schema-ren ikuspegi orokorra

Lengoaia grafikoei dagokienez, interfaze honek pantailan aurkeztu bezala definitzen du koda. Beraz, kode grafikoaren definizioa elementuen (aldagai, bloke, eta abarren) kokapenean (posizioan, forman eta tamainan) eta elementuen arteko loturen bideetan oinarritzen da. Interfazeak automatizazio proiektuan funtsezkoa izan daitekeen informazio osoa hartzen da kontuan. Informazio honetan ondoko hauek jasotzen dira: proiektuaren egitura (konfigurazioak eta konfigurazio bakoitzeko baliabideak), iruzkinak, programatzaileak definitutako datu motak, programatzaileak edozein lengoaia estandarren bidez sortutako funtzioak, funtzio-blokeak eta programak, eta baita aldagaiak eta atazen bidez antolatutako POU deiak dituen automatizazio-proiektua bera ere. Gainera, interfazeak mapaketa-informazioa eta fabrikatzailearen informazio espezifiko jasotzen ditu.

<sup>1</sup> [http://www.plcopen.org/pages/whats\\_new/tc6/status.htm](http://www.plcopen.org/pages/whats_new/tc6/status.htm) webgunean eskuragarri

4. irudiak PLCopen-eko XML *Schema*-ren ikuspegi orokorra zuhaitz egiturako itxura erabiliz aurkezten du. Bi programazio-ingurunek interfaze hau badute, informazio formatu honek beren arteko elkartruke zuzena bermatzen du. Formatu honen bidez elkartruka daitekeen informazioa IEC 61131-3 estandarrak definitutako 5 lengoaietako edozeinetan idatzia egon daitekeen POU bakar batetik automatizazio-proiektu oso bateraino doa. Honek guztiak, ordea, ez du bermatzen helmugako ingurunean proiektua errorerik gabe konpilatzea, batez ere fabrikatzailerek modu ezberdinak erabiltzen baitituzte IEC 61131-3 estandarra inplementatzeko. Aipatzekoa da, berriz, interfazeak berrerabilgarritasuna ahalbidetzen duela; hala ere, hardware menpekoek hardwarea bateragarria den kasuetan soilik funtzionatuko dute. Estandarrarekin bat datozen programazio-tresnetara softwarea egokitzeak, ordea, ez luke denbora askorik beharko.

## 5. ELKARREKINTZARAKO BIDEA

PLC-ak programatzeko merkatuan eskura daitezkeen ingurune gehienek aukerak eskaintzen dituzte kodea inportatzeko eta esportatzeko, baina batzuetan aukera mugatuak dira edota testuzko lengoaiekin lotuak. Kodea inportatzeko eta esportatzeko, PLCopen-eko XML formatua erabiltzen duten inguruneen artean aipatzekoa da, hala nola, *Phoenix Contact*-eko *PCWORX* tresna [11]. *Schneider*-eko *Unity Pro* [12] ingurunea bezalako beste batzuk, aldiz, PLCopen-ekoa ez den beste XML formatuez baliatzen dira. Badira inportatzeko eta esportatzeko beste testuzko formatu batzuk erabiltzen dituztenak ere, *Automation Alliance*-ko *CoDeSys* [13] tresna kasu. Azkenik, *Siemens*-eko *Simatic S7* [14] bezalako beste ingurune batzuk kodea gordetzeko formatu berezkoa erabiltzen dutenez, automatizazio-proiektuaren koderak API-az (*Application Program Interface*) baliatuz iristen da eta hortik abiatzen da esportazioa.

Elkarrekintzan jarri nahi diren bi inguruneek PLCopen-eko XML interfazea badute, elkarreragiketa zuzenean lortzen da. Aukera hau eskaintzen duten tresnak gero eta ugariagoak badira ere, asko dira oraindik horrelakorik ez dutenak. Atal honek jorratuk odu hain zuzen interfaze hori ez dagoen kasuetan elkarrekintzan nola lortu. Helburu hori dugularik, aplikatu beharreko teknologiak eta urratsak azaltzen dira hurrengo ataletan (ikus 5. irudia), beti ere PLCopen-ek proposatutako formatua oinarritzat hartuta.



**5. irudia.** Ingurune ezberdinak elkarrekintzan jartzeko bidea

### 5.1. Informazio-Teknologiak

Atal honetan, laburki deskribatzen dira W3C [8] delakoaren XML teknologiak elkarrekin erabiltzeko erabilgarriak gertatzen direnak. Hauek modu egokian erabiliz aplikatuz gero, informazioa ingurune batetik beste batera transferitzea ahalbidetzen dute.

- **XML Stylesheet Transformazio-Lengoaia (XSLT)** [15][16]:  
XML estilo-orri edo XSL-orri deitzen direnek (.XSL) ahalbidetzen dute plantilak edo *templates* direlakoak erabiliz XML dokumentuak iragaztea eta prozesatzea. XSL-n bi *template* mota daude aukeran. Lehenengo honek, *match* izenekoak, XML elementu jakin bati aplikatzeko prozesamendua jasotzen du. Prozesatze hau transformazioa egituratzen laguntzen duten bigarren *template* mota diren *name* izenekoak erabiliz ere antola daiteke. Bi *template* mota hauetako egitura 6. irudian adierazten da.  
Teknologia honen bidez automatizazio-proiektu bat esportatzen duen fitxategian fabrikatzailearen informazio espezifikoak gehitu daitezke. Proiektuaren kodea beste tresna batean inportatzean, aldiz, teknologia honi esker garrantzizko informazioa iragaz daiteke.

```
<xsl:template match="rootElementua">
<!-- algoritmoa -->
<xsl:apply-templates select="xmlElementua">
<xsl:with-param name="param1" select="param1Balioa"/>
<xsl:with-param name="param2" select="param2Balioa"/>
</xsl:apply-templates>
<!-- algoritmoa -->
</xsl:template>

<xsl:template name="Algoritmo1">
<xsl:param name="param1">
<xsl:param name="param2"/>
<!-- algoritmoa -->
</xsl:template>
```

**6. irudia.** XSL teknologiako *match* eta *name template* direlakoan adibideak

— **XML Interfazeak:**

*Document Object Model* (DOM) [17] eta XML-rako *Simple API* (SAX) [18] XML interfaze motak dira. Hauek beregainak dira programazio-lengoaiarekiko eta plataformarekiko. XML interfazeek programazio-lengoaia batekin batera, (Java, C++ edota C# esaterako) XML dokumentuak sortzeko eta manipulatzeko balio dute.

## 5.2. Elkarrekintzan jartzeko metodologia

PLCopen-eko XML interfazea duten bi inguruneren artean elkarreragiketa zuzenean lortzen dela jakinda, atal honetan azaltzen dira interfaze hori ez duten PLC-ak programatzeko ingurune ezberdinak elkarrekintzan jar ditzakeen metodologian eman beharreko urratsak. Bide hau, aurkeztutako XML teknologiak aplikatuz ibiltzen da (ikus 7. irudia):

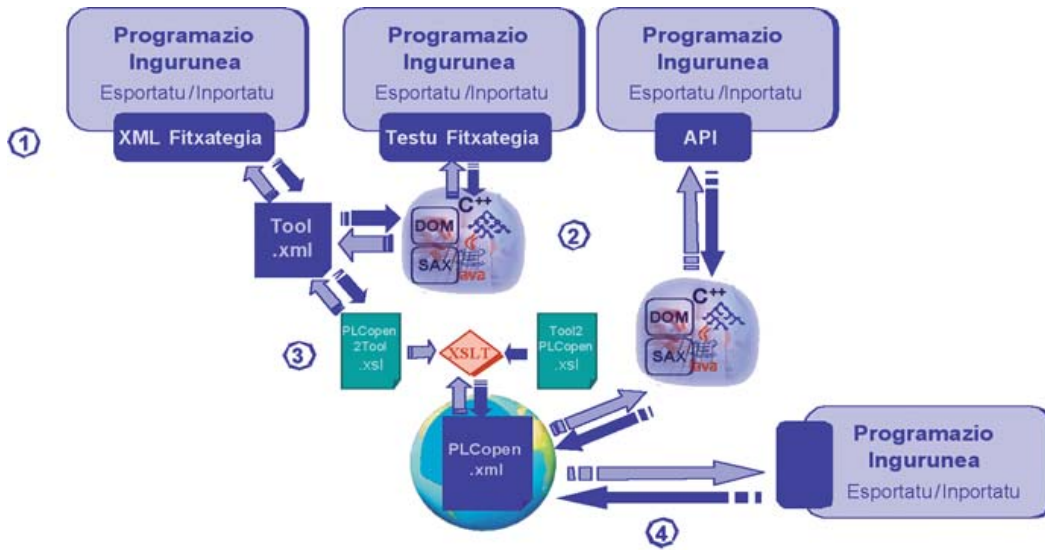
1. Proiektu osoa edo POU-a lehenengo ingurunetik esportatu
2. Esportatutako fitxategia prozesatu dagokion informazioa lortzeko
3. Fitxategi prozesatua PLCopen-eko XML formatuan adierazi
4. PLCopen-eko formatuko Proiektua/POU-a beste ingurune batetik inportatu

Hortaz, PLCopen-eko interfazerik gabeko inguruneetan, esportatu/inportatzeko erabiltzen duten modua kontuan hartuta, bidea ibiltzeko ondoko kasuak bereizten dira:

- PLCopen-eko interfazea ez den beste XML formatuan inportatu/esportatzen duten tresnak
- Testu egituratuko formatuan inportatu/esportatzen duten tresnak
- Kodea beste formatu batean gorde eta inportatu/esportatzeko API funtzioak eskaintzen dituzten tresnak

Elkarreragina ahalbidetzeko proposatutako metodologiak edozein ingurune PLCopen-eko formatura egokitzea du helburu. Ondoren, kasu bakoitzean jarraitu behar den prozesua azaltzen da.

PLCopen-ekoa ez den beste XML formatuan esportatzen duen tresnaren kasuan, esportatutako fitxategia (*Tool2PLCopen.xml*) XSL-orri egokiaren bidez prozesatzen da. Honela, dagokion informazioa iragazi eta PLCopen formatura egokitzeko bihurketa egiten da. Mota honetakoa inportatzen duen tresna denean, beste XSL-orri bat (*PLCopen2Tool.xml*) erabiltzen da PLCopen formatuko fitxategia tresnaren formatura egokitzeko.



**7. irudia:** Kontrol-programak trukatzeko kasu ezberdinen ikuspegi orokorra

Inportatzeko/esportatzeko ondoren azaltzen da jarraitu beharreko prozesua, ingurunea testu formatuko fitxategiez baliatzen denean. Alde batetik, esportatutako fitxategia, duen egiturari eutsiz, XML formatuan adierazten da. DOM edo SAX XML interfazeen bidez, testu-fitxategia XML fitxategi bihurtu eta behin hau lortuta, XML fitxategian XSL-orri egokia aplikatuz PLCopen-eko formatua lortzen da. Bestalde, aipatutako motako ingurunea inportatzen duen ingurunea denean, aldez aurretik PLCopen-eko formatuan adierazitako fitxategitik tresnaren gramatika erabiltzen duen XML fitxategia lortzen da, eta ondoren dagokion testu-fitxategia lortu inportatua izan dadin. Hau guztia, XSL-orriak, SAX eta DOM aplikatuz egiten da.

Informazioa beste edozein formatutan gordetzen duten inguruneetan inportatu/esportatzeko fitxategia SAX edo DOM-ekin batera API tresnak eskaintzen dituen funtzioak erabiliz egiten da. Hemen, berriro, esportatzea PLCopen formatua lortzean datza, eta inportatzea, aldiz, PLCopen formatuko fitxategia ingurunearen ezaugarrietara transformatzean datza. [6] artikuluan, xehetasunez zehazten da mota guztietako tresnetarako teknika hauen aplikazioa.

Aipatzekoa da metodo honek prozesuaren mantentze erraza bermatzen duela, ingurunearen eguneraketak edota PLCopen-eko formatuaren eguneraketak izanez gero, biak beregainki jasotzen baitira.

## 6. ONDORIOAK

Automatizazio arloan, IEC 61131-3 estandarrak PLC-ak programatzeko moduak bateratzeko aukera eskaini du. Bost programazio-lengoaiak definitzeaz gain zehazten duen softwarea antolatzeke ereduari POU izeneko unitateen bidez kodea modularizatzea ahalbidetzen denez, automatizazio-aplikazioaren programa osoaren edo zatien berrerabilera lor daiteke. Kodea beste programazio-tresna batean berrerabili ahal izateko aldeaz aurretik transferitu behar da. Proiektu osoen transferentziaren kasurako PLCopen erakundeak XML interfaze bat definitu du. Honela, PLCopen-eko XML interfazea erabiliz PLC-ak programatzeko inguruneen arteko elkarreragina lortu eta softwarearen benetako berrerabilera erdiesten da. Hau egia izatera iritsi da interfazeak aukera eskaintzen duelako pantailan dagoen informazioa beste ingurune batzuetara transferitzeko. Artikulu honek, gainera, azaldu du automatizazio aplikazio baten kodea nola inportatu/esportatu, PLC-ak programatzeko edozein ingurune izanik, eta PLCopen interfazea dutelarik edo ez. Hau lortzeko, hots, kodea benetan trukagarri bihurtzeko, aurkeztu den metodologia PLCopen-ek proposatutako XML formatuan oinarritu da. Metodo honek, ordea, aukera eskaintzen du kodearen berrerabilera baino zerbait garrantzitsuagoa lortzeko. Izan ere, proposatutako interfazeak beste software mota bateko tresnen artean ere elkarrekintza lortzeko erabil baitaitezke, besteak beste, konfigurazio, dokumentazio, bertsio kontrol, ereduak egiteko edota simulatzeko tresnen artean.

## ERREFERENTZIAK ETA ESTEKAK

- [1] MARTÍNEZ LASTRA, J. L. eta LÓPEZ OROZCO, O. J. 2004. «Estándares IEC 61131-3 e IEC 61499 (1.ª parte: Lenguajes Gráficos)». *Automática e Instrumentación*. **Bol. 353**. 67-76.
- [2] INTERNATIONAL ELECTROTECHNICAL COMMISSION. 2003. *IEC International Standard IEC 61131-3:2003. Programmable Controllers, Part 3: Programming Languages*.
- [3] PLCopen nazioarteko erakundearen Webgunea: <http://www.plcopen.org>
- [4] LEWIS, R.W. 1998. *Programming Industrial Control Systems using IEC 61131-3*. IEE Control Engineering Series.
- [5] JOHN, K-H. eta TIEGELKAMP, M. 2001. *IEC 1131-3: Programming Industrial Automation Systems*. Springer.
- [6] ESTÉVEZ, E., MARCOS, M. eta ORIVE, D. 2007. «Automatic generation of PLC automation projects from component-based models». *Int. Journal of Advanced Manufacturing Technology*. **Bol. 35**, Issue 6. 527-540. Springer.
- [7] PLCopen-eko TC6 komitearen XML interfazea: [http://www.plcopen.org/pages/tc6\\_xml/](http://www.plcopen.org/pages/tc6_xml/)
- [8] W3C. XML W3C Recommendation. Webgunea: <http://www.w3.org/TR/2006/REC-xml11-20060816/>

- [9] W3C. XML Schema. Webgunea: <http://www.w3.org/XML/schema>
- [10] VAN DER VLIST, E. 2002. *XML Schema*. O'Reilly.
- [11] Phoenix Contact. PC WORX. Webgunea: [http://www.phoenixcontact.com/automation/32131\\_31906.htm](http://www.phoenixcontact.com/automation/32131_31906.htm)
- [12] Schneider-Electric. Unity Pro. Webgunea: <http://www.schneider-electric.com>
- [13] CoDeSys Automation Alliance. Webgunea: <http://www.3s-software.com/>
- [14] Siemens Automation. Simatic S7. Webgunea: <http://www.automation.siemens.com/mcms/simatic-controller-software>
- [15] W3C. XML Stylesheet Language Transformation. Webgunea: <http://www.w3.org/TR/xslt>
- [16] TIDWELL, D. 2001. *XSLT*. O'Reilly.
- [17] W3C. DOM. Webgunea: <http://www.w3.org/DOM>
- [18] SAX. Webgunea: <http://www.saxproject.org>