This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Papakonstantinou, Nikolaos; Karttunen, Juha; Sierla, Seppo; Vyatkin, Valeriy
Design to automation continuum for industrial processes

Published in:
IEEE International Conference on Emerging Technology and Factory Automation, ETFA 2019

DOI:
[10.1109/ETFA.2019.8869325](https://doi.org/10.1109/ETFA.2019.8869325)

Published: 01/01/2019

Document Version
Peer reviewed version

Please cite the original version:
Papakonstantinou, N., Karttunen, J., Sierla, S., & Vyatkin, V. (2019). Design to automation continuum for industrial processes: ISO 15926 - IEC 61131 versus an industrial case. In *IEEE International Conference on Emerging Technology and Factory Automation, ETFA 2019* (pp. 1207-1212). [8869325] (Proceedings IEEE International Conference on Emerging Technologies and Factory Automation). IEEE.
<https://doi.org/10.1109/ETFA.2019.8869325>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Design to automation continuum for industrial processes: ISO 15926 – IEC 61131 versus an industrial case

Nikolaos Papakonstantinou¹, Juha Karttunen², Seppo Sierla³, Valeriy Vyatkin³

¹VTT Technical Research Centre of Finland Ltd, Espoo Finland, nikolaos.papakonstantinou@vtt.fi

²Outotec Oyj, Espoo, Finland, juha.karttunen@outotec.com

³Aalto University, Espoo, Finland, name.surname@aalto.fi

Abstract—Modern industrial processes have grown to be increasingly complex and their development is under tighter time and resource constraints. During the design lifecycle, the handover between process design and automation engineering is a critical step. This paper presents a standards-driven method for supporting the design to automation continuum. The source data format is ISO 15926 Proteus XML. Automation software describing control loops, alarms and interlocks are added as generic attributes in the Instrumentation Loop Function of the Proteus file. The target format is PLCopen XML for IEC 61131-3 based automation software. Control software code, global variables and the main program declarations and body, is auto-generated and defines the basic structure of the automation application. The standards-based method is compared with an applied approach based on state-of-the-art tools used by the industry.

Keywords—ISO 15926, DEXPI, Proteus, IEC 61131, PLCopen XML, process engineering, automation engineering

I. INTRODUCTION

The development process of a modern industrial system consists of several phases, starting from requirements engineering and functional modelling to detailed modelling, implementation and testing. A significant step in the development lifecycle is the data handover from process engineering to automation engineering.

This paper presents two approaches for addressing the design to automation handover challenge. Outotec, a Finnish engineering company, has developed a practical method based on a commonly used engineering database for transferring automation data to a Distributed Control System, trying to keep the data independent from the DCS vendor. The second approach is based on standard data formats for process design (ISO 15926) and automation software projects (IEC 61131). This effort demonstrates the possibility for adding basic automation information to the Proteus XML data container storing the Piping and Instrumentation Diagram and then using this information to generate the basic structure of the automation application based on the PLCopen XML data format for industrial automation software.

This paper presents a method based on the ISO 15926 and the IEC 61131 standards for enabling the first steps of basic control software engineering during the process design phase. This method is compared at a high level to an approach with a similar goal developed by an engineering company.

II. RELATED RESEARCH

A. CAE Vendor-driven solutions

CAE software vendors have identified the need for a holistic approach to complex system design. Their solutions involve the integration of the design tools to a common engineering database, creating a vendor-specific ecosystem.

Examples of this approach can be seen in the Catia platform by Dassault Systems [1], the COMOS platform by Siemens [2], the AVEVA Engineering by AVEVA [3] and the SmartPlant Foundation by Hexagon PPM [4].

While these implementations do provide practical solutions to industrial customers, their proprietary nature raises concerns related to data interoperability between vendors. Project migration is problematic, as well as the communication between project stakeholders who use CAE tools from different vendors.

B. Research methodologies based on standards

Machine-readable design artefacts are the basis for more productive engineering methodologies. Model driven engineering (MDE) refers to a broad category of methodologies that automate manual work related to translating the outputs of one phase in the design lifecycle to the inputs of another phase. Some researchers focus on standards to overcome the need for proprietary point-to-point solutions between tools from different vendors. AutomationML, which builds on the CAEX and PLCopen XML schemas, is a popular choice in the research community. [5] discusses the creation of metamodels from the said schemas and the use of Object Constraint Language (OCL) as a technology for enforcing constraints. [6] broadens the scope of the problem to consider a team of experts working on an integrated plant model, so that each of them needs to access and modify only a part of the model. [7] extends the MDE capabilities of AutomationML with a metamodel to support versioning and linking. [8] proposes a MDE solution for reconfiguring production systems, based on AutomationML and PLCopen XML. [9] combines MDE and AutomationML to generate modular control hardware and software designs from modular machine designs.

III. BRIDGING PROCESS AND AUTOMATION DESIGN

A. A practical approach to bridge process and automation design

A practical methodology for phase early control software engineering was developed by the engineering company Outotec to provide a solution to the process to software development gap. The goals for this development were to:

- improve plant engineering efficiency and quality and remove vendor locks.

- Achieve price and quality benefits due to the increased competition between DCS suppliers and reduce the migration costs due to DCS vendor change.
- Gain the ability to select the DCS vendor according to customer needs (each customer has typically different

DCS preference), which leads to increased customer satisfaction.

- decrease the required engineering hours, reduce the lead time of the project and reduce the need to start early DCS implementation before the process design is completed, because the software generation is very rapid.

- Be able to postpone the control software production until all preceding design phases are completed or at least the design is frozen. Most software re-work and multiple iterations can be avoided by implementing it right the first time.

- be able to apply modifications during the project implementation phase (incremental program generation, adoption of agile working methods) which leads to increased competitiveness due to the decreased need for programming resources.

- reduce the need of highly skilled human resources, since the DCS code can be (almost) programmer independent.

In this approach, the information of the process and instrumentation aspect of the plant are stored in an engineering database. This database is then customized by adding:

- Parameters used inside process loops. This data is enough to mass generate individual control loops (which are not yet connected anywhere).
- Connections between different loops implementing the DCS functional design (e.g. valve open/close commands, forced open/close commands and interlocks / permissives). This data is required in order to build a functioning DCS application.

The “Lean DCS” concept is introduced as part of this method. The basis of this concept is that the all the high-level control functions, that are not time-critical or depend on excessive data volume, are separated from the low-level vendor-specific control layer. The high-level control layer is DCS vendor-independent, and there is a clear interface to the low-level functions. The aim is that the software development focuses on the high-level control while the low-level services are auto-generated.

B. A standards-based method for the design to automation continuum

The second method for addressing the transition from process design to control software engineering utilizes standard data representations for process diagrams and control automation software.

The ISO 15926 standard, with the extension of the DEXPI specification and the Proteus XML file format was used for representing P&ID information while the IEC61131 PLCopen XML technology was chosen for storing the control software information.

Following the proposed methodology, the process topology and instrumentation is developed in a CAE tool that supports the ISO 15926 - DEXPI export format. After the P&ID development is done, the next step is the addition of the information related to the automation software. Ideally, the CAE design tool can be extended to facilitate the entry of this additional data while the user still has the overview of the P&ID. The end result is a Proteus XML file with the P&ID data and the control automation software information.

The control loop software information is stored in the corresponding Instrumentation Loop Function XML elements of the Proteus file in a separate Generic Attributes element with the value “AutomationAttributes” for the “Set” attribute.

Table 1 presents an initial proposal of new types to extend the DEXPI specification, which can then be mapped to control software elements in a PLCopen XML file. The purpose of this proposal is to provide a concrete basis to initiate discussion of extending the DEXPI specification with relevant stakeholders.

Table 1 Proposed mapping of information from DEXPI to PLCopen XML

Proposed new type	Usage in process design	Mapping to control software
ControlLogic Option	for defining the type of control (like on/off control)	local variable to the main program for representing the controller of the desired type
ControlLogic InputVar	the input measurement	global variable for the input measurement, assigned to the input of the controller
ControlLogic OutputVar	the control signal	global variable for the control signal assigned to the output of the controller
ControlLogic UpLimit/ControlLogic DownLimit	for setting up the controller (depends on the controller type)	input parameters to the controller
Highlimit Alarm/Lowlimit Alarm	for setting up alarms based on the measured input value	global variables for the alarm and the alarm threshold value; adds an IF-THEN statement to the program body to trigger the alarm
Highlimit Interlock/Lowlimit Interlock	for setting up cross-loop interlocks	global variable for the interlock threshold value; adds an IF-THEN statement to the program body to force a value to a control signal

A model to model transformation tool can parse this enhanced Proteus XML file and use the automation software attributes to create an internal model of the automation data. This information can then be used to generate a PLCopen XML file with the basic control loops, ready to be imported to a DCS tool that supports IEC 61131 PLCopen XML. A high level overview of the process is shown in Fig. 1.

IV. CASE STUDIES

A. Case study for the practical approach to connect process and automation design

While details on the actual project data related to this case cannot be shared due to confidentiality reasons, this section

Figure 1 – A high-level view of the model to model transformation. Some levels of hierarchy were removed due to space limitations.

will focus on the application of Outotec's method and the toolchain used.

The implementation of the method developed by Outotec is based on the AVEVA P&ID [10] and the AVEVA Engineering [3]. The P&ID design information is integrated with the engineering database provided by AVEVA Engineering database. The engineering database is extended to handle automation software information that can be directly used as source to generate control loops and interconnections between control loops in the vendor specific DCS system (in this case the Siemens SIMATIC PCS 7 DCS). The code generation from the database data to the DCS assumes the existence of libraries for the different controller types, sensors, actuators and indicators.

B. Case study for the standards-based method for process design to automation continuum

The standards-based approach is demonstrated with a case study based on the DEXPI reference P&ID example. This diagram presents a simple process where liquid flow generated by the P4711 pump is coming from the left, goes through the H1007 heat exchanger and then to the top of the T4750 tank. This flow can be manually operated (e.g. for maintenance reasons) with a hand switch (HS 4750.01). There is an outlet at the bottom of the T4750 tank. There are two return pipelines, one after the P4712 pump and one that goes through the H1008 heat exchanger to feed the final output line. The first return line is a safety loop. In case the pressure, indicated by the pressure indicator (PI 4712.01), is too high, some liquid flow will go back to the tank through the safety valve SV 104.1. The pressure of the output line (leaving the diagram to the right) is controlled with the pressure-indicating controller (PICSA 4712.02) and the fail close valve (PV 4712.02) that regulates the water flow back to the tank through the H1008 heat exchanger. The tank temperature is controlled in a cooling loop using a Temperature Indicator Controller (TICSA 4750.03) and a fail open valve (TV 4750.03), which controls the secondary flow in the H1008 heat exchanger. There are two control loops in this example process:

- Tank temperature control loop: The Temperature Indicator Controller (TICSA 4750.03) controls a fail open valve (TV 4750.03) regulating the secondary water flow in the H1008 heat exchanger.

- Output line pressure control loop: Pressure indicating controller (PICSA 4712.02) controls a fail close valve (PV 4712.02) that regulates the water flow back to the tank through the H1008 heat exchanger.

There is an alarm and an interlock related to these control loops:

- Alarm: If the tank temperature is high, then a high temperature alarm is triggered.
- Interlock: If the tank temperature is too high, then the PV 4712.02 opens fully, overriding the control commands (connection between temperature and pressure control loops).

Fig. 2 shows some generic attributes for the automation software related to setting up the temperature control loop, contained in the **InstrumentationLoopFunction** Proteus XML element. This information is enough to define the use of an on-off controller and set up its input, outputs and control parameters. These attributes also specify the high temperature alarm and the temperature to pressure control loop interlock. The exact format of these generic attribute elements is not important; the attribute sets of these elements are just an example to demonstrate the method. A formal specification needs to be developed with the contribution of all interested stakeholders in order to define what attributes are needed, where they need to be placed in the Proteus file and what parameters should these attributes contain.

This information is parsed by a prototype model to model transformation tool. Part of the resulting PLCopen XML file that contains the generated control software code is stored as Structured Text (ST) IEC 61131 code, as shown in Fig. 3. The auto-generated code (global variable definition and main program setup) assumes a library of controllers in the DCS environment and is based on injecting code in an empty PLCopen XML project file acting as a template.

The resulting PLCopen XML file is imported in the CODESYS DCS environment. In this simple example, the code is ready to run, even though there is no process simulation, which would be needed for testing and demonstration.

```
<GenericAttributes Number = "7" Set = "AutomationAttributes">
  <GenericAttribute Name = "ControlLogicOption" Value="temperatureControlTICSA" Type = "OnOffControlPOU"
  Tag="4750.03" Format = "string"/>
  <GenericAttribute Name = "ControlLogicInputVar" Value = "tankTempActual" Tag="4750.03" Type=
  "GlobalVarReal" Format = "string"/>
  <GenericAttribute Name = "ControlLogicOutputVar" Value = "valveControlTV" Tag="4750.03" Type=
  "GlobalVarBool" Format = "string"/>
```

Figure 2 – Part of the control software data for the temperature control loop in the Proteus XML export file of the DEXPI P&ID example. This information could be added using the CAE tool user interface, but in this case they were added manually to the XML file.

C. Comparing the two methodologies

The two methods focus on the same challenge, but they make a different optimization in the trade-off between standard support and immediate applicability. Outotec's approach, is based on customizations in a proprietary engineering database. It is successful in representing automation software attributes in a way that is directly usable by the Siemens Simatic DCS, but also potentially importable by DCSs from other vendors. The significant advantage of this method is that it is immediately applicable and successfully bridges the process design to automation development gap.

The second approach, based on the ISO 15926/DEXPI with its Proteus format and IEC 61131 with its PLCopen XML format, is not immediately applicable. The CAE vendor support for the Proteus format is still a work in progress. Some CAE vendors do support earlier versions of the Proteus format and it is expected that at least the software vendors participating to the DEXPI group [11] (e.g. Hexagon, Autodesk, Aveva, X-Visual) will eventually offer complete support to the ISO 15926/DEXPI standard. The support for PLCopen XML is at a better state with BECKHOFF TwinCAT 3 [12], CODESYS v3 [13] and MULTIPROG [14] being examples of development suites that support this data format. Since PLCopen XML is the data format adopted by AutomationML, there is a possibility that additional DCS vendors support it in the future.

These two approaches should converge in the future if vendor support for industrial data format standards is improved. This will remove the need for vendor-dependent customizations and ad-hoc solution development.

V. CONCLUSIONS AND FUTURE RESEARCH

Improving the handover between design lifecycle phases is a complex effort that requires interdisciplinary expertise and the availability of engineering tools with open interfaces.

This paper presented an industrially applied method independent from DCS vendor platforms while still based on a specific engineering database. This compromise proved to be good, since Outotec was able to adopt the method very fast and enjoy the benefits of the design to software continuum. There is the possibility for this method to become independent also from the design tool platform, when a standard information interoperability of process design is supported by the majority of CAE vendors.

An academic approach was also proposed, following standard data formats, based on the ISO 15926 and the IEC 61131. The potential of practical implementation of this second method is restricted by the small number of CAE and DCS vendors who support these standards. This approach was able to auto generate control loop code which was imported

by the target DCS system. There is potential for information from the P&ID graphics to be also used for the generation of the control panel HMI; this research direction is interesting future work.

This trade-off between immediate applicability vs long-term benefits is expected to continue to exist. Judging by initiatives like the DEXPI group, pressure for the adoption of data interoperability technologies and open interfaces will increase. Thus, there is hope that CAE vendors will be increasingly more open to technologies for interface and data schemas, which will bring benefits to all stakeholders involved in the lifecycle of complex industrial systems.

REFERENCES

- [1] Dassault Systèmes. (2019). CATIA V5-6R2016 drives higher Design excellence. Available: <https://www.3ds.com/products-services/catia/products/v5/portfolio/>
- [2] Siemens. COMOS – Making data work. Available: <https://new.siemens.com/global/en/products/automation/industry-software/plant-engineering-software-comos.html>
- [3] Aveva. Discover AVEVA Engineering. Available: https://www.aveva.com/en/Solutions/Product_Finder/AVEVA_Engineering/
- [4] Hexagon. SmartPlant Foundation. Available: <https://hexagonppm.com/products/asset-lifecycle-information-management/smartplant-foundation>
- [5] O. Kovalenko, M. Wimmer, M. Sabou, A. Lüder, F. J. Ekaputra, and S. Biffl, "Modeling AutomationML: Semantic Web technologies vs. Model-Driven Engineering," in 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), 2015, pp. 1-4.
- [6] R. Mordinyi, D. Winkler, F. J. Ekaputra, M. Wimmer, and S. Biffl, "Investigating model slicing capabilities on integrated plant models with AutomationML," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016, pp. 1-8.
- [7] S. Biffl, E. Maetzel, M. Wimmer, A. Lueder, and N. Schmidt, "Linking and versioning support for AutomationML: A model-driven engineering perspective," in 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), 2015, pp. 499-506.
- [8] E. Estévez, F. Pérez, D. Orive, and M. Marcos, "A novel approach for flexible automation production systems," in 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), 2017, pp. 695-699.
- [9] A. Armentia, E. Estévez, D. Orive, and M. Marcos, "A Tool Suite for Automatic Generation of Modular Machine Automation Projects," in 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), 2018, pp. 553-558.
- [10] Aveva. AVEVA P&ID: An Intelligent, AutoCAD-based P&ID Design Application. Available: https://www.aveva.com/en/Solutions/Product_Finder/AVEVA_PID/
- [11] DEXPI. Data Exchange in the Process Industry (DEXPI). Available: <http://www.dexpi.org/>
- [12] BECKHOFF. TwinCAT 3 - BECKHOFF New Automation Technology. Available: <http://www.beckhoff.com/Twincat3/>
- [13] CODESYS. CODESYS Development System. Available: <https://www.codesys.com/products/codesys-engineering/development-system.html>
- [14] PHOENIX CONTACT. KW Multiprog Pro v5.35. Available: <https://buy.advantech.eu/Automation-Software/SoftLogic-Software/model-MPROG-PRO535E.htm>

```
<body>
<ST>
  <xhtml xmlns="http://www.w3.org/1999/xhtml">pressureControlPICS471202.downLimit:=2.0;
pressureControlPICS471202.upLimit:=3.0;
pressureControlPICS471202.inputSignal:=gv1.outputPressActual471202;
gv1.valveControlPV471202:=pressureControlPICS471202.controlSignal;
temperatureControlTICS475003.downLimit:=30.0;
temperatureControlTICS475003.upLimit:=60.0;
temperatureControlTICS475003.inputSignal:=gv1.tankTempActual475003;
gv1.valveControlTV475003:=temperatureControlTICS475003.controlSignal;
IF (gv1.tankTempActual475003>gv1.highAlarmThreshold475003) THEN gv1.highAlarm475003:=TRUE; END_IF
IF (gv1.tankTempActual475003>gv1.interlockThreshold475003_471202) THEN gv1.valveControlPV471202:=TRUE; END_IF
```

Figure 3 – Part of the automation control software as generated in the PLCopen XML file (main program POU). The code is injected in an empty project template file previously exported by the CODESYS tool.