Abstract

Today’s manufacturing operations are faced with the need to reduce cost and be more competitive with as small resources as possible. Therefore, engineers need to use high level methodologies and software to facilitate the development, maintenance, and documentation of the production lines. These must include modeling, formal validation, prototype simulation and testing before building a real machine.

The aim of this paper is to describe methods to be used by small and medium size enterprises for the design of industrial automation systems and to develop a methodology and guidelines applicable to training of engineers.

Keywords

IEC 61131, PLCopen, XML, IEC 61499, IEC 62424, CAEX, AutomationML, digital factory.

1 Introduction

The rise of global economy has generated a stronger competition for the manufacturing enterprise in every part of the globe. Therefore, manufacturers are facing numerous challenges across all phases of the product development process – shortened development times, managing global supply chains, fierce competition, and increasingly complex products. They are in need of solutions that can help them get to market faster, cheaper and with greater functionality and capability.

The best strategy is to adopt new development methods and open industrial standards for providing complete integration of multiple systems at the hardware and software level as well as to choose the latest available software tools that can have long term usefulness and can be used for at least five to ten years [1].

Below some functionalities and solutions for automation, which are under investigation, are listed:

- the reuse of existing automation system models in the development process, for teaching to use and maintain automated systems and for creating Human Machine Interfaces [1,2];
- usage of reconfiguration agents for manufacturing systems, which allows automatic reconfiguration of a manufacturing system for selected products (it takes care that the needed machines are available for production, downloads required programs and reconfigures the machines) [3];
- usage of neutral data formats for exchanging data between different automation engineering tools (it covers information about automation system structure (topology and geometry) and behaviour (logic and kinematics)) [4];
- usage of neutral data formats for exchanging data between technology equipment and their control systems [5];
- usage of the data modelling standards, for example, XML developed by world wide web standard consortium, data modeling and integration tools, for example, xmlspy, mapforce, developed by firm Altova. Such tools enable fast and seamless integration of manufacturing solutions and development of design methods.

The aim of this paper is to describe methods that can also be used by small and medium size enterprises (SMEs) for design of industrial automation systems and to develop a methodology and guidelines to be used for training of engineers.

In the next section an overview and analyses of data integration and design methods in the field of industrial automation are presented. The third part of the paper proposes some training methods and topics that cover requirement specification, modelling and prototype testing. The provided guidelines are for students and engineers working in SMEs.
2 Existing Design Methods

The newest design methods can be found in the existing and accomplished standards listed below. These are used by SME engineers and also in larger companies to develop modern industrial automation systems.

- IEC 61499 – open standard for distributed control and automation;
- IEC 61131 – standard for programming Programmable Logic Controllers (PLC) for industrial applications;
- IEC 62424 – standard for representation of process control engineering;
- AutomationML – open data format for a digital factory.

2.1 IEC 61131

The IEC 61131 standard is the first standardized programming languages for industrial automation [6], previously called IEC 1131.

The IEC 61131 standard provides a methodology for making control software models for Industrial Process Measurement and Control Systems (IPMCSs) [7]. The standard defines a software model for defining industrial automation projects as well as 5 programming languages (like instruction list, structured text, ladder diagram, function block diagram, and sequential function chart) for making control software for PLC. By the decomposition of industrial automation systems into logical elements and modularization, the control software can be structured into small programs for increasing program code re-usability, reducing errors and increasing software efficiency. This standard assumes only one PLC to control an industrial automation system.

Today IEC 61131 standard (part 3) is promoted by the standard organization PLCopen. The members of that organization are developing a data format, which allows one to exchange existing PLC programs, libraries and projects between different control software development environments without losing information [8]. The data format is based on eXtended Markup Language (XML) and it is called PLCopen XML. All control software information is stored as sequential function chart, even the corresponding I/O-relations [4].

2.2 IEC 61499

The IEC 61499 standard has been developed especially as a methodology for modelling open distributed IPMCSs to obtain a “vendor-independent” system architecture. The standard has an advanced form IEC 61131, because the previous one did not allow one to use a decentralized controlling system in automation, in addition, it did not show connections between distributed function blocks and how to reuse these function blocks in other applications.

This standard defines concepts and models so that control software (which defines the behaviour of an industrial automated system) is encapsulated in function blocks, which can be assembled and distributed to controller nodes, which are spatially distributed [9]. This means that the control software has been divided between different devices and each device controls a small part of the automated system (like in a car washing system a pressure sensor regulates a liquid valve so that the liquid pressure is kept on same level during the washing operation). The IEC 61499 standard provides a very good reference model based on function blocks which can be applied and reused for the design and engineering of large scale distributed industrial automation systems. So it allows one to develop object oriented control software for control applications in the industrial automation [10]. The IEC 61499 function blocks look different from the IEC 61131 function blocks with a head, or top half, for event input and output connections. Rightly or wrongly, all IEC 61499 function blocks are event driven. The bottom half has the inputs and outputs of defined data types. Function blocks come in types: basic function blocks, composite function blocks, and service interface function blocks. For function block programming the languages described by the IEC 61131 standard but also high programming languages like Java and C# can be used.

2.3 IEC 62424

The IEC 62424 standard defines procedures and specifications for the exchange of Process Control Engineering (PCE) relevant data provided by the Piping and Instrumentation Diagram (P&ID, process visualization) tool [11]. The standard defines a neutral data format CAEX (Computer Aided Engineering Exchange) for integrating PCE tools and P&ID tools together (storing and exchanging information between these tools). CAEX is an abstract object oriented data format based on XML, which depicts real or logical plant objects in the form of data objects [5]. CAEX combines model-techniques with meta-model-techniques. The model-techniques allow storage of object information that is common across different vendors, e.g. objects, attributes, interfaces, hierarchies, references, libraries, and classes. The meta-model-techniques allow a flexible definition of object information that is usually individual and application dependent, e.g. certain attribute names, specific classes or object catalogues. So CAEX can be applied to all types of static object information, e.g. plant topologies, document topologies, product topologies, petri nets, but also for non-technical applications like phylogenetic trees [12, 13].

Modifications of the plant are no longer only caused by an engineering fault, but increasingly enforced by evolution and ongoing adaptivity and interoperability of production systems. The abstract data exchange format can be used for description and automatic configuration of production monitoring and control systems. Such data formats are a key to the idea of “automation of automation”.

89
2.4 AutomationML

AutomationML (Automation Markup Language) is a neutral data format based on XML for the storage and exchange of engineering information about industrial automation systems (plant) [1]. It covers information about the plant structure (topology, geometry and kinematics) and the plant behaviour (logic) [4].

AutomationML adopts several specialized standards under one umbrella to support as many aspects of the engineering process as possible. It incorporates different standards through strongly typed links across the formats (see Fig. 1):

- **CAEX (IEC 62424)** for describing industrial automation system topology as a hierarchical structure of the objects used in the automation system (their properties and relations). This structure is the AutomationML top level format.
- **COLLADA™** for describing 3D geometry of the objects, physical interconnection of 3D objects and their geometric dependencies to support motion planning. This information is stored in a separate XML file.
- **PLCopen XML** for describing the behaviour of objects and the automation system in a sequence of actions including the corresponding I/O-relations and logical variables.

![Fig. 1. Basic architecture of AutomationML [4]](image)

The goal of AutomationML is to interconnect the heterogeneous automation tools of modern engineering production planning of the different disciplines, e.g. mechanical plant engineering, electrical design, visualization development, PLC, and robot control. This means that different companies can work on one automation system together even if they use different software tools.

The methodology behind AutomationML helps to specify and structure the objects of the industrial automation system (main and sub objects), make 3D models of objects, specify the behaviour of the object, make control software models and test the overall system virtually before the real system has been built. The methodology simplifies the usage of library files and reduces automation system specification efforts.

3 Modeling and Testing Using CIROS

Before models are created, engineers and trainers should have logical ideas about the automated systems. The next stage is to model these ideas.

Models have a very important part in developing industrial automation systems. Virtual models show how factory automation systems are integrated, what they look like and how they work before building a real system. Earlier the models were made of mathematical formulas and 2D graphics, which were understandable only by highly trained specialists. Thanks to 3D software, like Autodesk’s 3ds Max and SolidWorks, it is possible to make 3D models of the mechanical parts of the automation system and animate their work. But the animation is not seamlessly connected to system control software.

![Fig. 2. Assembled factory model in CIROS](image)

Now there are special engineering tools called digital factory tools like RobotStudio, CIROS and DELMIA [14]. CIROS software contains all the necessary components for the implementation of the Digital Factory, and for the integrated engineering of production lines ranging from the planning and modeling stage, including programming and control, up to the simulation, optimization and the virtual commissioning. It is possible to make 3D virtual models of entire industrial automation systems, for example, a full factory (see Fig. 2). The modules of a factory can be assembled in the CIROS development environment and then their work can be simulated in real time. These modules can be made with different engineering tools (see Fig. 3),
which are provided by different firms. Then the full system can be tested and optimized according to specified criteria that can be mechanical, logical and functional. Functional requirements are specified, for example, in PLC and robot programs. The digital factory tools can be effectively used for teaching and training of students, engineers and end-users, for example, operators and maintenance people.

The integration of the CIROS digital factory tool with AutomationML is still complicated, because CIROS environment does not support strongly XML data formats. For this a special integration tool is needed. Thus, a large amount of engineering work has to be carried out for this purpose.

The newest solution for teaching is realized by Festo in the iFactory concept. The iFactory has a systematic modular construction, enabling students or engineers to try out new ideas immediately. All iFactory production cells are equipped with topology feedback so that the digital factory tools and the reconﬁguration agent automatically recognizes the constructed production line. All system settings and conﬁgurations are generated automatically while arranging and connecting the iFactory cells [15]. The price of such a learning factory starts from 1 million EUR.

4 Analysis and discussion

The methodologies described in the standards are ﬁrstly reﬁned and used by standardization organization members. Mostly they are the leading companies and universities in their ﬁeld. They are trying to harmonize automation system design methodologies in order to achieve better collaboration and compatibility between members but also some competitive advantage.

In order to involve engineers and designers from SMEs into the technology transfer process some effort is needed from the universities side. They can promote discussion groups and additionally provide top level practical seminars or training for groups of engineers from interested SMEs.

One of the important topics is introduction to standards and application programming. The most widely used standard is IEC 61131 (most PLC manufacturers point out that their engineering tools support it), but it does not support a decentralized controlling system (distributed automation system). Therefore, IEC 61499 standard has been developed, but currently only few software engineering tools are supporting it. Such advanced tools are used for making distributed automation system control software. Distributed control systems can be implemented not only in factory engineering but also in the field of power engineering. Such a future power system is called a “Smart grid”.

Besides control software it is necessary to introduce engineering tools for modeling of industrial automation systems. These models can be developed and used for testing and optimizing the system. The philosophy is implemented in AutomationML and in the following tools: CIROS, simFMS [16], RobotStudio (for making virtual models of manufacturing lines, where ABB robots are used, programming and simulating their work). These tools help to describe the automation system topology, objects, geometry, kinematics and the behaviour of the system. The IEC 62424 standard can be used for topology description. The digital factory training can be based on AutomationML, but it has not yet been completed and is not widely used in training facilities.

Last but not least, the problem is how to implement the new methodologies and technologies faster. There exist numerous training sites that already use complex didactic solutions which are produced by companies who develop didactic solutions. A possibility proposed in this paper is to start building “bridges” between data model structures that are implemented in training software packages and the data models that are planned to be implemented in AutomationML. For this purpose the data modelling and integration tools, for example, XmlSpy, mapforce developed by ﬁrm Altova, can be used. As result of integration it is easier to start transfer of different model data between widely used didactic solutions. This allows more efﬁcient combining of training and real engineering work in a factory.

5 Conclusion

No general architecture which allows seamless engineering and training in the ﬁeld of industrial automation is available today.

The promising basic architecture for automation engineering is AutomationML, which stores information about topology, geometry, kinematics, logic, references and relations in order to close existing gaps during the design of industrial automation systems.

In the ﬁrst stage the engineers of SME should take iFactory training courses offered by training facilities, which use advanced modeling and simulation tools like CIROS. After learning modeling, programming and simulation, engineers can integrate the knowledge obtained into the designing process of a real industrial automation system. They also need to explore or continue to follow technology trends like AutomationML to be better prepared for new digital factory tools based on emerging XML technologies.

Acknowledgement

This work was supported by ESF DoRa program activity 6, funded by European Social Fund and managed by Archimedes Foundation.
References


