The Journal of Systems and Software 84 (2011) 1100–1113

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Model-driven development of industrial process control applications

David Hästbacka*, Timo Vepsäläinen, Seppo Kuikka

Tampere University of Technology, Department of Automation Science and Engineering, P.O. Box 692, Korkeakoulunkatu 3, FI-33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 26 February 2010
Received in revised form
20 December 2010
Accepted 30 January 2011
Available online 1 March 2011

Keywords:
Model-driven development
Domain-specific modeling
UML
Profiles
Process control applications

ABSTRACT

This article presents model-driven development and domain-specific modeling applied to the development of industrial process control applications. The approach is based on established practices of the industrial automation and control domain, a compatible UML profile, and an integrated and standards based tool environment for modeling and transformation execution. The methods provide means for developing applications using domain-specific modeling concepts to increase productivity and enhance platform independent solution reuse. The approach has been implemented to support industrial practices and to be able to utilize existing control system platforms. During demonstrations and an assessment event with industry professionals the methods have been successfully applied to the development of small-scale process control applications. In this paper, attention is also paid on discussion of the practical applicability and benefits of the approach for engineering and development of industrial process control applications.

1. Introduction

New demands originating from business requirements have become more complex for industrial process control applications. This is due to upper level manufacturing execution (MES[1]) information content and functionality as well as mission critical reliability, availability, maintenance, safety and security (RAMS) issues. At the same time, the application developer organizations have been forced to cut down the development times to market. Therefore, both the expressiveness and the efficiency of software development for control applications need to increase.

Challenges within the development of control applications have also increased. The application development is a complex undertaking where persons with multidisciplinary backgrounds need to collaborate. Input information for requirements is acquired mainly from process engineers in various formats (piping and instrumentation diagrams (P&ID[2]), spreadsheets, and XML documents conforming to domain specific schemata). On the other hand, runtime environments for application execution are various. They include proprietary distributed control system (DCS) platforms, advanced programmable logic controller (PLC) platforms as well as networked, desktop, and embedded environments. Therefore, domain specific, but platform independent modeling of applications is needed.

The development process should provide automatic transfer and – where appropriate – automatic model transformations between modeling phases. Domain specific requirements, design and implementation concepts on different levels of abstraction need to be properly aligned. Modeling elements, understandable for process and control system engineers alike, are needed to express the requirements on application design originating from process engineering. Modeling concepts and methods are also needed to enable specification of various control structures and loops as well as control algorithms (proportional–integrative–derivative, fuzzy, model predictive, etc.) with domain-specific concepts. Also instrumentation (sensors, actuators) has to be modeled on a proper level of application abstraction. The modeling concepts should promote fluent information exchange and understanding between different design disciplines, and with compatible concepts, also enable automatic transfer of otherwise error-prone design data handover.

AUKOTON is a research project started in the beginning of 2008 aiming to define concepts and methods to enhance process control application design. The project is carried out by Finnish research institutions in co-operation with multinational companies representing the process industry, the energy and sustainability sector, engineering offices and industrial control system vendors. The process builds on existing domain specific knowledge, standardization efforts and standardized application data formats, while taking into account current development practices and the technology platforms used by the industry. In the AUKOTON project our focus

* Corresponding author. Tel.: +358 40 849 0016; fax: +358 33 115 2340.
E-mail addresses: david.hastbacka@tut.fi, david.hastbacka@gmail.com (D. Hästbacka), timo.vepsalainen@tut.fi (T. Vepsäläinen), seppo.kuikka@tut.fi (S. Kuikka).

[1] Manufacturing Execution Systems is the layer of information systems concerning planning, monitoring and control of manufacturing processes in the space between enterprise level systems and control systems of the physical process.

[2] A schematical diagram presenting the piping and process flow between equipment and instrumentation typical to process plants.

0164-1212/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2011.01.063

in Tampere University of Technology (TUT) has been on model-driven development (MDD) concepts and further development of the UML profile, developing tool support for the process, and providing guidance for adopting new methods to current practices. The MDD approach developed has also been evaluated in an organized hands-on event by a few professional designers from the participating companies. In this assessment event the participants used the developed methods and tools to design a small-scale process control application. Feedback was gathered using problem notes and interviews between different phases in the development process.

This paper is organized as follows. Section 2 presents related work from the domain of automation and control and software engineering in general. Section 3 presents some of the challenges in currents practices developing control applications for industrial processes. The conceptual approach is presented in Section 4 detailing the concepts and the transfer of information during development of process control applications. Focal issues and applicability is discussed in Section 5 and the paper is concluded with future work in Section 6.

2. Related work

MDD of large-scale control applications has not been studied to the extent comparable to many other domains and areas, such as application development for embedded systems, for example. Domain-specific modeling and applications for some specific areas of use have been studied, however, but without much notice to related engineering disciplines or existing practices typical to design of automated manufacturing plants. There is also no domain-specific language (DSL) for the automation domain that is adequate for modeling control functionality compatible with standard methods and existing practices that can be extended to more than only some special application area. The AUKOTON approach discussed in this paper utilizes the UML Automation Profile (UML AP) modeling concepts, the preliminary version of which was previously also developed at TUT (Ritala and Kuikka, 2007), and currently has been enhanced, implemented and given tool support.

The need to increase the efficiency of control system development without compromising quality has been acknowledged by Karaila and Systä (2007) among others. In the referenced approach, template meta-programming techniques were applied to a graphical function block programming language and type circuits, typical to DCS application programming, to increase productivity and solution reuse in practical projects. According to Karaila and Systä (2007), projects that use templates to a full extent can implement most of the application programs using templates and a specific template library. In spite of a high degree of reuse the approach is unfortunately restricted to a vendor specific control system platform (Karaila, 2010).

In the domain of automation of manufacturing plants the AutomationML (Drath et al., 2008) is an interesting effort striving to provide a neutral data format for exchange of design data between different engineering tools and the separated engineering phases. It provides concepts to store information about plant topology, geometry, kinematics, logic, and other references and relations using existing standards, e.g. CAEX, COLLADA and PLCopen XML, and object orientation to encapsulate different aspects of engineering. The AutomationML approach specifies how to transfer different types of information, thus, also restricting the solution space by supporting technologies on a relatively detailed level close to the few PLC based implementation platforms supporting the PLCopen XML format.

UML based code generation has been an interesting subject also in the domain of automation and control. UML behavior diagrams, for example, have been used as a visual programming language for a production control system based on agents (Kohler et al., 2000). Code generation from UML diagrams is challenging as it depends on the concepts being modeled, the designers point of view and often vague semantics. In the referred approach, however, the authors have restricted the UML notation to a subset of the language that has precise semantics. As a result, executable (Java) code is generated from general UML concepts. More recently UML has also been proposed to describe a model of a DCS and control requirements on a high level of abstraction (Basile et al., 2008). In this case, code generation is only used for program skeletons (C++). Both of the referenced approaches rely on standard UML and provide merely guidelines on how UML can be applied.

The use of UML with distributed control systems and IEC 61131 (IEC, 2003) has been analyzed by Bonfe and Fantuzzi (2003, 2004), and Chiron and Kouiss (2007) among others. Witsch and Vogel-Heuser (2009) have presented an approach for integrating object-oriented modeling using UML to IEC 61131 development. Vogel-Heuser et al. (2005) have previously proposed a method for generating IEC 61131-3 (ST and SFC) code from general UML models. Lu et al. (2005) have proposed a similar approach but with more emphasis on separating platform independent and platform specific models similar to the AUKOTON approach. Both of these referenced code generation methods share similarities with AUKOTON in that platform specific features are tagged as stereotypes for inclusion of platform components in the generation process of executable applications.

Ramos-Hernandez et al. (2005) have proposed an environment for developing distributed control systems using IDN (Integrated Design Notation) for bridging gaps between software engineering using UML, process control execution using IEC 61131-3 and control engineering using Simulink. Development of distributed systems using UML has also been studied in the FLEXICON project (Marcos et al., 2004). The method defines the UML Specification profile used to design the application functionality and the IEC_Profile for characterizing IEC 61131 standard elements. The models are used in accordance to the OMG (Object Management Group) MDA (Model-driven Architecture) (OMG, 2003) initiative to finally produce code with use of a code-generator subsystem. The FLEXICON tool set has further been expanded to simulation of distributed systems in applications such as aerospace, marine, automotive and process control (Thompson et al., 2007).

IEC 61499 (IEC, 2005) is a standard for developing distributed control and automation applications based on the function block concept. MDD of distributed control applications using UML and IEC 61499 has been studied by Thramboulidis et al. (2007) and tool support has been developed for the approach to allow the use of UML and UML profile extensions such as stereotypes and tagged values in the analysis stage of the development process (Tranoris and Thramboulidis, 2006). According to a previous study (Thramboulidis, 2004) the best alternative for using UML for development of control applications is a UML profile that is tailored for control and automation domain.

UML has also been proposed as a solution to describe the system considering the whole development process using transformations from UML models to IEC 61499 models (Panjaitan and Frey, 2007; Hussain and Frey, 2006). Moreover, UML has been studied for modeling IEC 61499 applications to avoid misinterpretation of models and to resolve possible compliance issues between the model and the implementation on diverse run-time platforms (Hussain and Frey, 2007). Vyatkin et al. (2005, 2009) and Grabmair et al. (2007) have discussed object-oriented modeling and the use of UML with IEC 61499 and the function block architecture. Dubinin et al. (2005) have proposed a framework where UML is combined with IEC 61499 to support engineering. In the referenced approach, UML class, sequence, cooperation and state chart diagrams are used to model the structure, dynamics, component connections and com-