SysML-Based Approach for Automation Software Development – Explorative Usability Evaluation of the Provided Notation

Daniel Schütz, Martin Obermeier, and Birgit Vogel-Heuser

Chair for Automation and Information Systems
Technische Universität München, Boltzmannstraße 15,
D-85748 Garching near Munich, Germany
{schuetz,obermeier,vogel-heuser}@ais.mw.tum.de

Abstract. The rising complexity of production automation systems and especially their automation software require new engineering concepts to support their development. Model-based concepts for the interdisciplinary development of production systems have been proposed in many research projects. In previous works of the authors an approach to enable interdisciplinary development of automation software, based on the Systems Modeling Language (SysML), has been developed and evaluated. One of the most important notations for the developed modeling approach is the SysML Parametric Diagram (PD). This paper briefly introduces the newly adapted modeling notation of the PD and presents the evaluation results from empirical usability experiments with human subjects.

Keywords: Programmable Logic Controllers (PLC), Systems Modeling Language (SysML), Usability, Programming languages.

1 Introduction

The software complexity of modern production systems increases with growing market dynamics, forcing industrial enterprises to ensure the flexibility of production systems [1]. During the engineering process and the installation of a production plant, not only automation software engineers, but also a great variety of engineers from different disciplines (especially mechanical engineers) are involved. Unforeseen c onstraints on the plant design regularly require adaptations of the automation software during the installation of a plant, as adaptations of its mechanical design are often not possible. Hence, a common understanding of the automation software by the software engineer and mechanical engineer is required to support the cooperative search for a solution to realize the necessary adaptations.

During the research project KREAagentuse [2], a tool supported method has been realized that uses a model-based approach to support application engineers in handling software complexity. Inside the project, a model-based development approach for automation software based on the Systems Modeling Language (SysML) [3] and methods for applying this architecture for different automation software projects have

A. Marcus (Ed.): DUXU/HCII 2013, Part IV, LNCS 8015, pp. 568–574, 2013.

been developed. For modeling applications of the developed software architecture, editors for SysML diagrams have been integrated inside a standard development environment for automation software development conform to IEC 61131-3, which is the leading standard for programming of Programmable Logic Controllers (PLC). Among others, an editor for SysML Parametric Diagrams (PD) has been implemented, as the PD is the most important diagram to model the developed architecture [2]. Furthermore, a code generator that enables the direct implementation and execution of automation software modeled in PDs was developed.

In order to evaluate the usability of the PD and its editor a study was designed regarding its suitability for software and mechanical engineers in comparison to the languages of the IEC 61131-3. In this study, enabling an interdisciplinary development and the understandability of software in the proposed SysML modeling notation have been investigated. The study was developed in context of several previous studies. In their work Patig [4] and Gemino and Wand [5] present results of various experiments on the usability of modeling notations. Patig focuses on data modeling respectively on meta-model usability. Gemino and Wand further include diagram creation as a constructive (design) task; however, the use of UML for diagram creation is not included. In a previous research study two approaches based on UML and Idiomatic Control Language (ICL) [6] were compared to the standard procedure of PLC programming with IEC 61131-3 [7]. In this study, modeling the process prior to programming did result in significant benefits, although no significant differences between the used modeling notations resulted (cf. [8], [9]). It was remarkable that despite the additional modeling task and the participants' insecurities about how to create a model, time and error rates did not significantly differ from the state-of-the-art procedure of PLC programming without any modeling support. Further studies comparing UML and IEC 61131-3 [10] show that both languages can be learned with similar training effort.

This paper comprises the usability evaluation experiments and results conducted during the research project KREAagentuse. The remainder is as follows: The next section introduces the SysML Parametric Diagram as one of the newly developed programming notations as well as the legacy programming notations, which are used for the comparative study. The experimental design and evaluation results are presented in sections 3 and 4. Section 5 summarizes the paper and gives an outlook on future works.

2 Investigated Notations for PLC Programming

As shown by Gemino and Wand in [5] empirical studies on usability are an established scientific method. Hence, in order to investigate the usability of the SysML PD used in the newly developed programming approach, an empirical study has been conducted. The study was an experiment with one factor (notation) which was varied experimentally. The primary objective was to investigate the comprehensibility of model contents comparing the SysML PD to two legacy PLC programming notations, i.e. the Continuous Function Chart (CFC) and Structured Text (ST). In the following subsections, the considered notations PD, CFC and ST are introduced.

2.1 The SysML Parametric Diagram

Inside the Parametric Diagram of SysML a (software) module can be modeled by its parameters, sub-modules, and the connections between them. Therefore different language elements of SysML and corresponding notation elements of the PD (cf. Fig. 1) can be used. The parameters (software variables) of a module are represented using the language element ValueProperty. Sub-modules inside a PD are represented by either Blocks or ConstraintBlocks. In contrast to Blocks which can be further decomposed, ConstraintBlocks refer to a simple algebraic rule (ConstraintBlock).

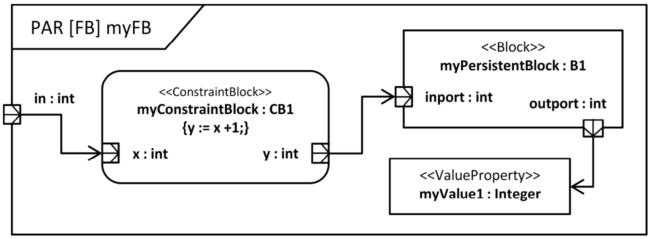

Fig. 1. Exemplary SysML Parametric Diagram as adapted by the newly developed approach

Blocks as well as ConstraintBlocks contain Ports that represent the data interfaces of the modeled (sub-)modules. These ports can be interconnected by BindingConnections (arrows) to define the data-flow of the modeled software.

2.2 PLC Programming Notations CFC and ST

The control notations ST and CFC are widespread and accepted in the field of PLC programming. ST is a textual, procedural programming language (cf. Fig 2., top) and part of the IEC 61131-3 standard [11]. CFC is a graphical PLC programming language that is most similar to the PD and widespread in the process industry as well as in the control technology. Although CFC is not part of the IEC 61131-3 standard it is a common extension of IEC 61131-3 programming environments.

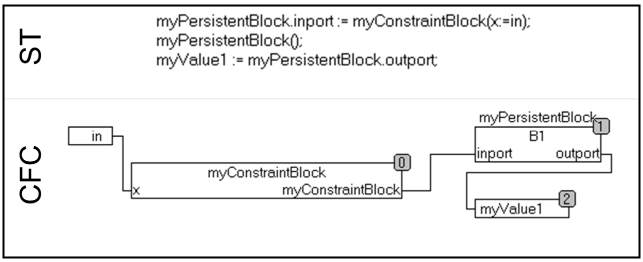

Fig. 2. Examples of considered legacy notations for PLC programming (top: ST, bottom: CFC)

The languages ST and CFC were chosen based on their prevalence and similar semantic significance to the PD. One of the major differences between the PD and the legacy notations CFC and ST is that the elements of a code model (e.g. decomposable and non-decomposable sub-modules) cannot be as easily distinguished. A disadvantage of the textual language ST compared to the graphical languages PD and CFC is that the data flow of the modeled software cannot be as easily reconstructed.

3 Design of the Usability Experiments

The design of the experiments follows the recommendations of Patig [4] and is based on the criteria of Parsons and Cole [12]: The experiment compares notations with the same information content not requiring any problem solving tasks (just model interpretation) and the participants are no experts (i.e. engineering students).

In order to obtain meaningful results in a comparative study all possible confounding variables should be measured or excluded. Questionnaires were developed to measure previous knowledge, the motivation and cognitive demand as well as to gain demographic data. Previous knowledge is tested by a set of eight multiple choice questions per notation on the content of a given model. This results in a percentage of correct answers for each notation. In order to provide a similar base level of knowledge in all notations a consistent training was designed. To exclude tool effects it is an appropriate method to conduct an experiment paper-based as Friedrich [8] describes. So our experiment design focuses on paper-based experimental tasks.

Table 1. Design of the usability Experiments

Time in Min	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
5	Previous Knowledge Test ST					
10	Training ST					
5	Previous Knowledge Test PD					
10	Training PD					
5	Previous Knowledge Test CFC					
10	Training CFC					
15	Break					
10	Task A CFC	Task A CFC	Task A PD	Task A PD	Task A ST	Task A ST
10	Task B ST	Task B PD	Task B ST	Task B CFC	Task B PD	Task B CFC
10	Task C PD	Task C ST	Task C CFC	Task C ST	Task C CFC	Task C (PD)
5	Questionaire on motivation, cognitive demand					

For the experiment the processing of three tasks was required. Task A, task B and task C were designed with similar complexity, in order to gain comparable results.

For the experimentation these three tasks were permuted in the notation sequence (cf. Table 1) in order to eliminate learning effects. Each task demanded the participants to interpret a given model created in CFC, PD or ST, showing relationships of variables and functions for different systems. In order to solve each task six questions on the contents of the given model (one multiple choice and five free choice questions) had to be answered. These ranged from identifying the type of given information to describing the signal flow from input to output variables through various functions. The experiment was planned with six participants (N=6, students of mechanical engineering) each in one cell (C1-C6), as shown in Table 1.

4 Results of the Usability Experiments

The experiment was conducted from October until November 2011 at the Technical University in Munich with students of mechanical engineering (N=6, one female, av. age = 23.33). The previous knowledge test results (scale 0-100% correctly solved questions) show comparable low mean values for the PD and ST notation ($m_{PD\_P} = 19.44\%$, $m_{ST\_P} = 23.61\%$), previous knowledge in CFC was at a lower level ($m_{CFC\_P} = 7.69\%$).

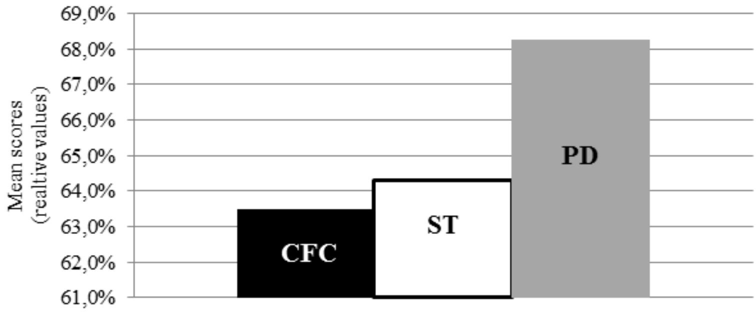

Fig. 3. Results of experimental tasks (scale 0-100%, i.e. percentage of correct answers)

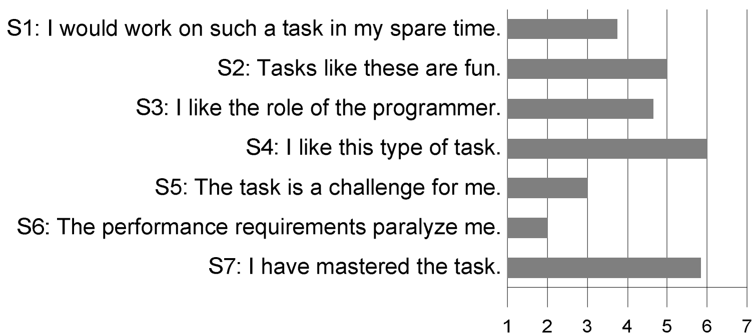

Fig. 4. Descriptive results of the questionnaire on motivation and cognitive demand (scale: 1 statement not fitting – 7 statement fully fitting)

The results of the experimental tasks in CFC, ST and PD, as shown in Fig. 2, indicate a positive tendency for the suitability of the PD as the PD tasks were solved best ($m_{PD} = 68.25\%$) with a positive offset of 3.97% to ST ($m_{ST} = 64.28\%$) and 4.76% to CFC ($m_{CFC} = 63.49\%$). No floor or ceiling effects occurred indicating a suitable level of task complexion.

The participants' answers in the final questionnaire give a subjective view on the cognitive demand and motivational aspects, as depicted in Fig. 3. The participants' motivation levels (S1-S4) span from a medium to an elevated level (cf. Fig. 4). Also the cognitive demand was rated medium to easy (S5-S6) and in conclusion the participants expected to have mastered the task (S7). These results are overall in order with the objective task performances.

5 Summary and Outlook

During the research project KREAagentuse [2], a prototypical tool support for implementing Programmable Logic Controllers (PLC) has been developed that uses adapted diagrams and notations of Systems Modeling Language (SysML). In this paper, the usability experiments that have been conducted to evaluate the adapted SysML Parametric Diagram (PD) as programming notation for PLCs were presented.

The usability experiments empirically compared the comprehensibility of the PD to the comprehensibility of two legacy programming notations, i.e. Continuous Function Chart (CFC) and Structured Text (ST), for human subjects (students from mechanical engineering). Despite the comparable previous knowledge in PD and ST, the PD tasks were solved best. This shows that the participants comprehended PD models after a short training at a comparable or even better level than the models in state of the art notations (cf. Table 1, Fig. 3). Therefore the results of the experiments indicate that SysML PD can be rated well-suited for the modeling of PLC software.

Further studies will investigate the usability of the PD for engineers which are experienced in the use of the legacy programming languages. Current projects are integrating the developed SysML programming notation with a development method for distributed automation systems. Therefore, future works will also investigate the applicability and benefit realized by supporting engineers with both, the development method and programming (modeling) language.

References

1. Westkämper, E.: Adaptable production structures. In: Manufacturing Technologies for Machines of the Future (21st Century Technologies), pp. 87–120. Springer (2003)
2. Frank, U., Papenfort, J., Schütz, D.: Real-time capable software agents on IEC 61131 systems – Developing a tool supported method. In: Proceedings of the 18th IFAC World Congress, Milan, Italy (2011)
3. Object Management Group: Systems Modeling Language (OMG SysML) V1.1 (2008)
4. Patig, S.: A practical guide to testing the understandability of notations. In: Proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling, vol. 79, pp. 49–58 (2008)

5. Gemino, A., Wand, Y.: A framework for empirical evaluation of conceptual modeling techniques. Requirements Engineering 9(4), 248–260 (2004)
6. Bristol, E.: Not a batch language; A control language! ISA Transactions 34(4), 387–403 (1995)
7. Friedrich, D., Pantförder, D., Vogel-Heuser, B.: Evaluation of UML in Process Automation- Results of an Experimental Approach. In: Presented at the 10th IFAC,IFIP, IFORS,IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC HMS), Seoul (2007)
8. Friedrich, D.: Applicability of conventional software engineering methods and tools for modeling and programming of control systems. Kassel University Press, Kassel (2009)
9. Braun, S., Obermeier, M., Vogel-Heuser, B.: Usability evaluation of modeling notations for software engineering in machine and plant automation. In: Proc. Emerging Technologies and Factories and Factory Automation (ETFA), Bilbao, pp. 1–8 (2010)
10. Vogel-Heuser, B., Obermeier, M., Braun, S., Sommer, K., Jobst, F., Schweizer, K.: Evaluation of Teaching an UML-based Software Engineering Approach Compared to IEC for PLC Programming. In: IEEE Transactions on Education, pp. 61131–61133 (accepted paper)
11. German Commission for Electrical, Electronic and Information Technologies of DIN and VDE, Programmable Controllers - Part 3: Programming languages. English version (IEC 65B/725/CD:2009). DIN German Institute for Standardization (December 2009)
12. Parsons, J., Cole, L.: What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. Data & Knowledge Engineering 55(3), 327–342 (2005)