Control Engineering Practice 21 (2013) 218–230

Contents lists available at SciVerse ScienceDirect

Control Engineering Practice

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

Automated generation of simulation models for control code tests

Mike Barth [a,*], Alexander Fay [b]

[a] ABB Corporate Research Center Germany, Wallstadter Straße 59, 68526 Ladenburg, Germany
[b] Helmut-Schmidt-University, Institute of Automation Technology, Holstenhofweg 85, 22043 Hamburg, Germany

ARTICLE INFO

Article history:
Received 12 March 2012
Accepted 28 September 2012
Available online 17 November 2012

Keywords:
Simulation
Model generation
Factory acceptance test
Modelica
CAEX
P&I diagram

ABSTRACT

The correct configuration of the control code is a critical part of every process control system engineering project. To ensure the conformity of the implemented control functions with the customer's specifications, test activities, e.g., the factory acceptance test (FAT), are conducted in every control engineering project. For the past several years, control code tests have increasingly been carried out on simulation models to increase test coverage and timeliness. Despite the advantages that simulation methods offer, the manual effort for generating an applicable simulation model is still high. To reduce this effort, an automated model generation is proposed in this paper. The models automatically generated by this approach provide a modeling level of detail that matches the requirements for the tests of the control code on the base automation level. Therefore, these models do not need to be as detailed as the high-fidelity models which are used for, e.g., model predictive control (MPC) applications. Within this paper, the authors describe an approach to automatically generate simulation models for control code tests based on given computer aided engineering (CAE) planning documents.

1. Introduction

1.1. Aim and scope of the automatically generated models

Process control engineering for automated production plants in the process industries can be divided into two main tasks: (a) the design and implementation and simulation-based test of the base automation control code, such as binary and continuous (mostly P or PI) control functions, sequences, interlocks, and the setting of limits for alarms and events, and (b) the design, implementation and tuning of advanced process control (APC) functions, including model predictive control (MPC) applications. Both, (a) and (b), require models of the plant, but both have different requirements on the level of detail of their models. The engineering of the base automation control functions requires a lower modeling level of detail than the design and implementation of APC. Whereas APC models require highly detailed modeled plant sections, e.g., a distillation column or a controlled heating system for a batch process tank, the simulation models needed for the test of the base automation include whole plant areas on a lower modeling level. These simulation models do not need to be appropriate for simulating every detail of a chemical process, but have to model mainly the qualitative effect, e.g., of a closing valve on its respective mass flow or the rising fluid level in a vessel in combination with the according signal of a level sensor, or the increase of heat energy due to the opening of a hot water valve. These tests belong to the group of open-loop control tests, including interlock logic tests as well as the testing of sequences. Such base automation control functions account for the majority of the control functions of a plant and therefore for a significant amount of the test effort. The authors explicitly exclude closed-loop tests such as the optimization of controller parameters because those tests would require a more detailed plant model. The aimed test cases will be further explained in Section 4. The approach described in this paper focuses on the automatic generation of the plant models required for the test of the base automation control functions.

1.2. Challenges during the engineering of process control systems

Today's process control engineers face various challenges: On one hand, the process control systems (PCS) become high performance products with a highly integrated functionality, which requires thorough consideration during design and installation. On the other hand, the engineers face continuously rising demands regarding the engineering projects. This includes the need to significantly reduce project durations as well as a rising need to be cost effective in comparison to market competitors. Both are trends which are in conflict with the overall aim to guarantee a good quality of the engineering results. This is

* Corresponding author. Tel./fax: +49 6203 716253.
E-mail address: mike.barth@de.abb.com (M. Barth).

0967-0661/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.conengprac.2012.09.022

Activity		Possibility of error	Effects of errors	Perceptibility of errors	Error probability	
1.1	Set project goals	0.6	1	0.6	0.36	X
1.2	Make a rough cost estimate	0.4	0.6	0.2	0.05	
2.1	Create a concept for the plant	1	1	0.8	0.80	X
2.2	Make a cost estimate	0.6	0.6	0.2	0.07	
3.1	Determine PCT functions	0.8	0.8	0.6	0.38	X
3.2	Collect process engineering data	0.8	0.6	0.4	0.19	
3.3	Determine technical implementation	0.8	1	0.4	0.32	X
3.4	Calculate costs	0.6	0.8	0.4	0.19	
4.1	Select equipment	0.6	0.8	0.6	0.29	X
4.2	Determine central facilities	0.4	0.8	0.6	0.19	
4.3	Specify a control system	0.6	0.6	0.8	0.29	X
4.4	Create loop diagrams	0.4	0.2	0.4	0.03	
4.5	Create loop function sheets	0.6	0.4	0.6	0.14	
4.6	Prepare installation documents	0.6	0.6	0.4	0.14	
5.1	Make arrangements for order placing	0.6	0.8	0.4	0.19	
5.2	Confirm delivery	0.4	0.2	0.2	0.02	
5.3	Configure software	1	0.6	0.6	0.36	X

Fig.1. Probabilities of process control engineering activities (NAMUR Worksheet 35, 2003).

important because most process plants operate within high temperature ranges, pressure ranges, and sometimes handle toxic media. Therefore, errors in the control code, e.g., wrongly set limit values or missing interlock logic, may cause damage not only to machines or the environment, but also to employees working inside the plant and to inhabitants of the surrounding area. Therefore, extensive tests of the process control system have to be carried out on site during commissioning. Almost 70% of the errors detected during the final commissioning of the process control system are software errors (Spath & Landwehr, 2001). Examples of such software errors are (according to Weule, Spath, & Schelberg, 1994):

- Multiple use of the same variable.
- Wrongly setting and resetting of variables.
- Typing errors.
- Missing or wrongly implemented interlock logic.

The international user association of automation technology in process industries (NAMUR[1]) has published a worksheet for the structuring of process control system projects (NAMUR Worksheet 35, 2003). As shown in Fig. 1, this worksheet categorizes engineering activities with regard to (a) the possibility of errors, (b) the severity of the effects of errors as well as (c) the difficulty of perception of errors. In regard to the aforementioned control code errors, the activity "Configure Software" (row 5.3) is characterized by a wide range of possible errors (value: 1.0) in combination with a high effect as well as a high difficulty of perception value (each: 0.6). Therefore, the resulting error probability (value: 0.36) of this activity is quite high. Even if these errors can be identified before the process plant is taken into operation ("hot commissioning"), the late on-side debugging tends to be expensive. The risk that not all errors are found will be increased by the pressure of the upcoming start-of-production date. Regarding the explained trends and analyzed figures, there is a strong need to identify as many control code errors as possible within an early engineering phase.

[1] www.namur.de

1.3. Test activities within the engineering of process control systems

A variety of methods exist to ensure the quality of the control code. The most common methods are verification and validation. Verification is based on the availability of formal specifications. The term "formal" refers to a specification which is "strictly composed as well as syntactically and semantically well defined" (Frey, 2002). Examples for formal specifications are Petri Nets or Automata which can be used in combination with formal verification methods, e.g., reachability analysis or (symbolic) model checking. Formal verification enables to mathematically prove the correctness of the control code implementation (Dotoli, Fanti, Mangini, & Ukovich, 2011; Soliman & Frey, 2011).

However, in engineering projects in the process industries, the specifications for the control functions are usually provided in terms of informal documents (Frey, 2002), e.g., timing diagrams, text documents, P&IDs (piping and instrumentation diagrams), etc. Therefore, verification would require the formalization of the informal specification and would thus lead to additional efforts. Therefore, informal validation – e.g., testing – has become the most common method to check (without mathematical proof) the correctness of the control code.

Testing can be categorized into the methods of Black-Box-Testing and White-Box-Testing. Whereas White-Box-Testing allows the PCS engineer to see the internal code structure, Black-Box-Testing is used to test the automation system without having knowledge of the control code itself (Frey, 2002). In this context, Black-Box-Testing aims to provide inputs to the tested system and measures the output (system answer). An example for a long-established practical implementation of this testing method is the use of so-called "switch boards". A switch board consists of electrical switches and potentiometers which are used to stimulate the necessary test signals in the process control system. Today the classic electrical wired switch boards are replaced by software applications, which mark the transition to simulation-based testing methods. Because this paper focuses on the testing of compiled control code that has already been uploaded to the controller, Black-Box-Testing needs to be applied.

As explained in this section, the area of control code testing marks a critical engineering phase regarding the quality of the