INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

APPROACH FOR SCENARIO-BASED TEST SPECIFICATIONS FOR VIRTUAL COMMISSIONING

M. Schneider, J. Gausemeier, T. Schmüdderrich and A. Trächtler

Keywords: production system development, test specifications, virtual commissioning, simulation-based design

1. Introduction

In a turbulent, highly complex environment companies need to plan their production systems quickly and accurately without time-consuming and cost-intensive iteration loops to achieve competitive advantages. However, this is restricted by the increasing complexity of such systems. This mainly depends on various requirements of interdisciplinary development. In particular, the performance of modern mechatronic products is affected by a close interaction of mechanics, electronics, control theory and software engineering. The development process combines the aspect of the different disciplines. The same applies to production systems (e.g. dough production systems).

In order to achieve the planning objective, it is still necessary to survey the entire development process. This is one of the reasons why Systems Engineering (SE) has been developed. Systems Engineering is a continuous, interdisciplinary approach for the development of complex technical systems. This approach foregrounds the entire system and focuses all development activities and the associated project as well (Figure 1). SE keeps also socio-economic issues in mind [INCOSE 2010], [HNI et al. 2013].

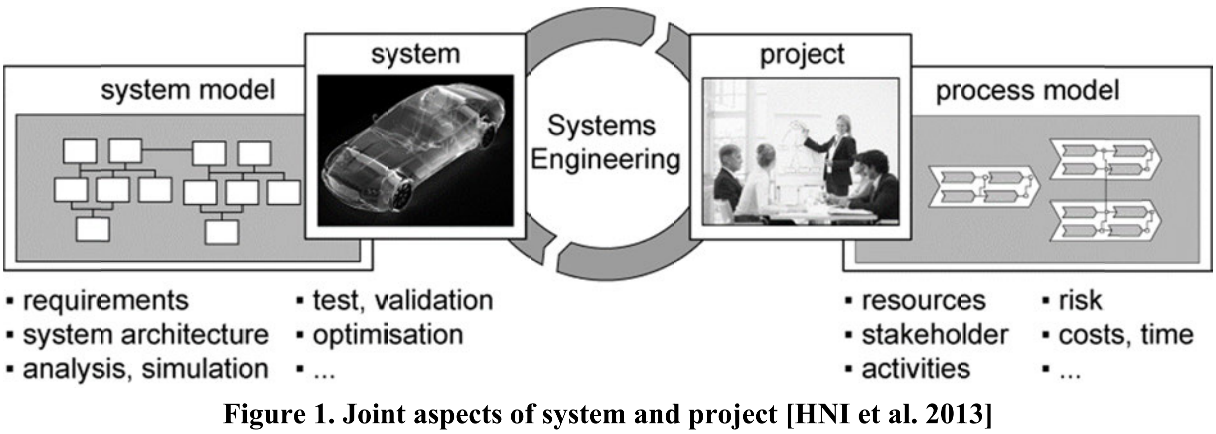

Figure 1. Joint aspects of system and project [HNI et al. 2013]

Common procedure models and techniques that support the planning process are developed to help companies plan their production systems [Ehrlenspiel 2007]. The VDI guideline 2206 takes the V-model as a basis of design methodology for mechatronic systems [VDI2206], whereas the VDI guideline 2221 provides a general-purpose, sector-independent approach to the development and design of technical systems and products [VDI2221].

Due to the multidisciplinarity of advanced production systems, models are widely used to simulate the system behaviour in early development phases. The primary objective of virtual commissioning (VC) is the validation of the system's control program [Wünsch 2008], [Sauer 2011]. The lifecycle of production systems usually begins with the conceptual design via the detail design, the manufacturing and assembly, the operational phase and finally ends with the redistribution [Schmüdderrich et al. 2013]. The transition from the planning phase to the operational phase is called production ramp-up. During ramp-up, the commissioning (C) ensures the readiness for production.

Consequently, commissioning is one of the most important steps in the machinery and plant engineering [Wiendahl et al. 2009]. In terms of Systems Engineering virtual commissioning is a supportive tool for developing, testing and validating product-associated production systems. The conformance to requirements and error correction can be ensured by this approach (Figure 1). In this manner, interdisciplinary development can be successfully completed within a reasonable period of time [HNI et al. 2013].

Virtual commissioning comprises the steps modelling, realisation and evaluation to simulate the behaviour while planning and setting up the production system. The basic principle includes a simulation model of the system that is linked to a control model or to the real control. This system model is composed of mechanical, hydraulical, pneumatic and electrical components [Wünsch 2008]. Creating behavioural models for the virtual commissioning usually necessitates a temporal effort. Passing the steps of virtual commissioning will lead to time saving and enables an earlier start of production [Wünsch 2008], [Schmüdderrich et al. 2013]. Thus, it seems reasonable to check the effort against the benefit of the model (Figure 2).

In this paper, we focus on scenario-based test specifications for virtual commissioning to simulate expected system behaviour as well as unexpected behaviour in case of failure. The first section analyses the challenges for commissioning, in particular the problem of testing the system behaviour with regard to virtual commissioning. Secondly, the demonstrator (a monorail conveying system for manufacturing and logistical processes) used to illustrate the test specifications is presented. Finally, our approach for scenario-based test specifications for the virtual commissioning is introduced initially, before it is demonstrated.

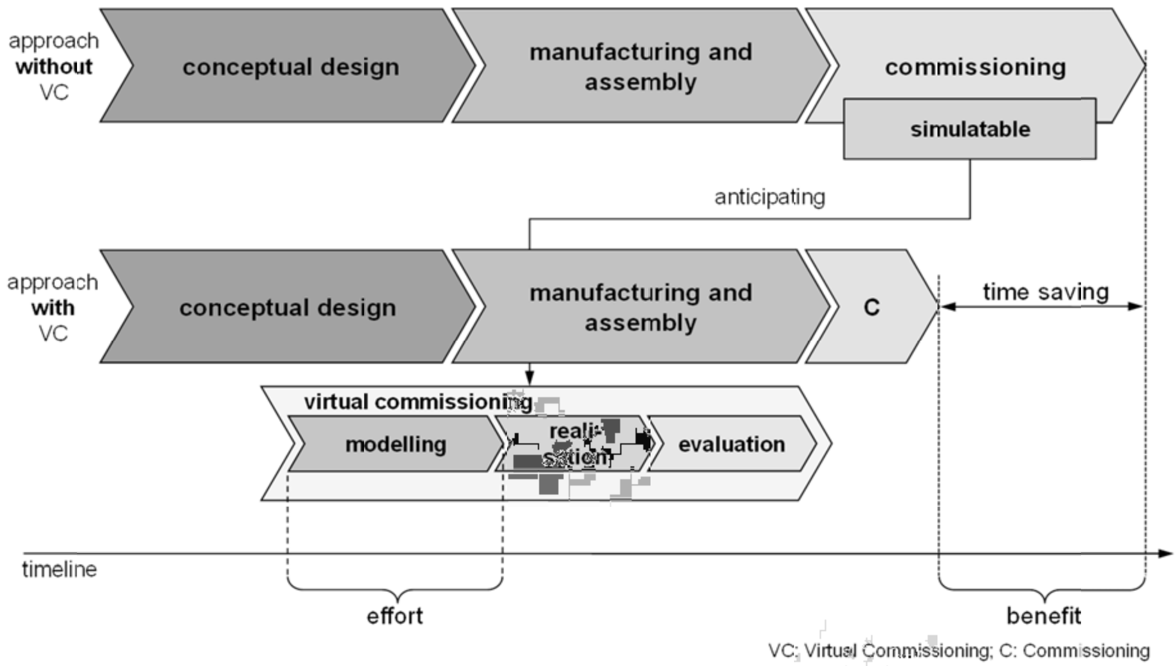

Figure 2. Basic principle of virtual commissioning [Wünsch et al. 2005]

2. Challenges for commissioning

The following section illustrates the challenges for commissioning and covers the issue of the demand for a methodical test specifications for the purpose of virtual commissioning. The commissioning, as already mentioned, is one of the most important steps. Its main objective is the function readiness of assembled single components consisting of hardware (e.g. mechanic, hydraulic components) and software (e.g. control program) components. Furthermore, the commissioning ensures the correctness of the overall function, because 80 % of bugs and failures are caused by insufficient planning and development [Masing 2007], [Wünsch 2008].

Due to the growing percentage of control engineering in advanced production systems, the proportion of software in the overall system increased considerably in the last years. Comparing the duration needed for hardware development, software development and commissioning of the years 1980 and 2009, it is obvious that the time exposure for software development and commissioning significantly increased. In contrast the time exposure of hardware development decreased. During commissioning electronic and programmable logic controller (PLC) components consumes the main part of the entire time. Hereof, the major part is correcting software errors. This is complicated by the fact, that some parts of software can only be on-site programmed [VDW 1997], [Wünsch 2008].

For function validation, different simulation methods are established. These methods distinguish between a real and a virtual level as well as between the controller (e.g. programmable logic controller) and the system (e.g. a plant). In the early stages of development, neither the control nor the system exist in the real world. Therefore, a model-in-the-loop (MIL) simulation is performed for the investigation of the principle solution at this point. Here, both the system and the controller exist as a model. If the control code of the real control is already available, it can be tested as part of a software-in-the-loop (SIL) simulation with the system model. To test the function of the real controller, all inputs and outputs of the real control system must be connected to the system model. This method is known as hardware-in-the-loop (HIL) simulation. The last method uses the real system and the simulated control and is called rapid control prototyping (RCP) [Isermann 2006]. The conjunction of real controls with real systems enables the most accurate findings. However, this option is not always available for various reasons. For example, during the simultaneous engineering of the system and the associated programmable logical controller, the system is not yet ready for software-testing. In many cases, established methods do not consider specific application scenarios as well as failure scenarios [Zäh et al. 2006], [Wagner et al. 2008].

Managing the quality or the risk of systems are methods and tasks of Systems Engineering. The generic term quality management comprises various guidelines or methods for quality assurance to detect failures at an early stage. Holistic strategies like Total Quality Management (TQM), Quality Function Deployment (QFD), Failure Mode and Effect Analysis (FMEA) and also Failure Tree Analysis (FTA) are exemplary approaches. Failures are not the only result of mistakes during design or programming faults but also arises from disturbances [Pahl et al. 2007].

In this paper, we present a scenario-based approach for test specifications for virtual commissioning to cope with the mentioned challenges during the development process. Virtual commissioning enables the development of the control program at an early stage of the engineering process by using behavioural models. Errors in the control program are often discovered only during operation. Due to our approach takes expected and unexpected behaviour into account. The sooner a test in the development process is conducted, the better errors could be detected and resolved [Wagner et al. 2008]. Test specifications regarding to application scenarios as well as failure scenarios can considerably reduce commissioning time and simultaneously increase the quality of the control program caused by an early error detection and correction. As a consequence the cost benefit is directly induced [Zäh et al. 2006].

3. Monorail conveying system

The approach for test specifications for virtual commissioning considered here, is demonstrated by an intelligent monorail conveying system for manufacturing and logistical processes. The conveying system with its components and functionality as the major demonstrator for this approach is specified firstly followed by an illustration of attached processing stations and robots.

The conveying system, which is illustrated in Figure 3 links a material store, a turning center, a milling center and a fully-automated assembly station. One gantry robot and two one jointed-arm robots execute the required handling tasks.
The monorail conveying system consists of four basic components: linear tracks, curves, switches and shuttles, which can be individually mounted according to task. In addition, the system has intelligent routing modules to control track sections as well as shuttles. Positioning units to approach an exact shuttle position are also components of the system. The shuttles are equipped with electronic drives, distance sensors and radio-frequency identification (RFID) chip. Information to identify the shuttle can be saved on the chip, for example the shuttle number and state of the current workpiece.
Relating to the different processing stations five workpiece states can be distinguished: *empty*, *unmachined*, *assembled*, *milled* and *turned*. The shuttle is tagged as empty if no workpiece is loaded. Unmachined means that the shuttle has received a raw workpiece, whereas milled and turned mark if the process has been conducted. Assembled means the shuttle carries finished workpieces. Those interactions result in several challenges to be solved for the control program and the required commissioning.

Figure 3. Overview of the monorail conveying system with proper processing stations and robots

4. Approach for test specifications for the virtual commissioning

This section addresses the issue of a scenario-based test specifications for the virtual commissioning. For this purpose, the approach is separated into two types of scenarios to verify the control program (Figure 4). On the one hand the approach comprises application scenarios and on the other hand failure scenarios. Application scenarios specify in which preferred modality a system should behave in operation situations. Additionally, special operation situations are also included in application scenarios, e.g. maintenance or manual mode. In contrast, failure scenarios specify in which preferred modality a system should behave in unexpected operation situations. In case of a failure, a safe state must be guaranteed. The control program as well as the behaviour models need to be developed with regard to both types of scenarios and their resulting requirements, respectively. Then, the control program can be verified and analysed in the course of virtual commissioning and optimised if necessary.

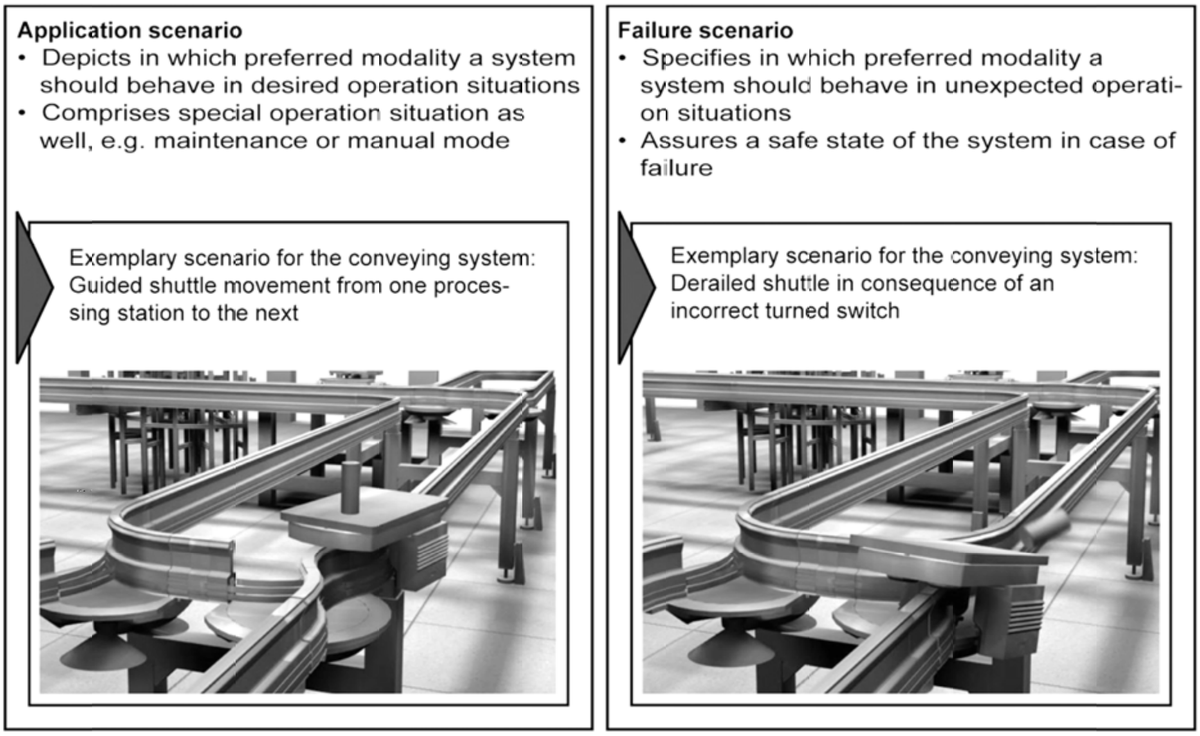

Figure 4. Types of test scenarios for virtual commissioning

The following sections illustrate the procedure for test specifications based on application scenarios and failure scenarios, respectively. Therefore, the types are firstly illustrated in general followed by an integrated example of the intelligent monorail conveying system as described in section three.

4.1 Application scenarios

To verify the system with regard to the application scenarios, requirements for the control program and the models representing these scenarios need to be deduced. To do so, operation and special operation situation need to be identified, analysed and evaluated. For this reason, the application scenario comprises four steps (Figure 5). The first step is to identify application scenarios for representative operation and special operation situations to define the desired system behaviour in such situations. The procedural model for the virtual commissioning on the basis of model-based design described in [Schmüdderrich et al. 2013] can be used for this purpose. Especially the conceptual design phase is concerned with application scenarios. Analysing the entire system life cycle may be helpful to identify different possible scenarios.

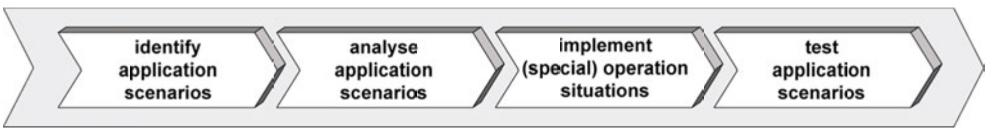

Figure 5. Procedure for application scenarios for test specifications

In addition to the holistic consideration of the system life cycle, each step that a workpiece passes through in the later production sequence needs to be analysed to achieve a maximum completeness of possible scenarios. However, the control program as the objective focus of virtual commissioning should be kept clearly in mind.

Subsequently, an analysis of the relevant application scenarios needs to be conducted in step two. Functional modules for each application scenario need to be determined. The active structure specifies functional modules, their attributes and interactions between further functional modules. At first, each functional module needs to be itemised. Due to this, involved functional modules can be identified and allocated to operation, special operation situation and application scenarios, respectively

[Schmüdderrich et al. 2013]. As a result of this the developer is enabled to select participating functions or related functional modules.
After selecting functions and functional modules, an implementation of the operation and special operation situations follows in the third step. Hence, the functions must be assigned to requirements, which they need to fulfil. The behavioral models also need to be adjusted in order to meet the requirements and thus represent the functions. Common methods or techniques for requirements engineering and management are described in [Hood et al. 2005], [Rupp et al. 2007]. These methods support the developer in deduce requirements with regard to virtual commissioning and as a result to improve control program.
Finally, the control program has to be tested and optimised according to the representative application scenarios on the basis of the established outcomes. Depending on the respective step in the development process, suitable methods to simulate the behaviour of the control program need to be chosen, e.g. model-in-the-loop, software-in-the-loop and hardware-in-the-loop or rapid control prototyping.
After all previous steps have been completed the test specifications for the considered application scenario is prepared in detail. Subsequently, further application scenarios have to be specified according to the same procedure.

4.2 Failure scenarios

In case of failure, the system must guarantee a safe state. To achieve this, the control program has to be verified with regard to unexpected situations, in particular failure and bugs. Hereby, the desired system behaviour in case of failure can be tested before the commissioning ensures the readiness for production. This section describes failure scenarios in the field of virtual commissioning in detail.
Firstly, failures have to be identified (Figure 6). In order to reveal all conceivable failures a methodical approach including a complete system and function analysis is recommended. Elements of failure mode and effect analysis (FMEA) and fault tree analysis (FTA) as described in [DIN25424 1990], [Pahl et al. 2007], [Brüggemann et al. 2012] can be used for this purpose. Similar to the identification of application scenarios, the production process steps may provide clues of failure. The FMEA and FTA as well as an analysis of the production process steps offers the opportunity to reveal potential of weak spots.
In the second step errors that could occur need to be analysed in detail. Therefore, the failures as well as their causes and consequences should be identified.
Subsequently, the causes have to be connected to related functional modules. Some errors result technical fault. For this reason, it is hard to test technical faults with the aid of virtual commissioning. Nonetheless, technical faults induce consequences and the control program has to switch the system to a desired safe state.

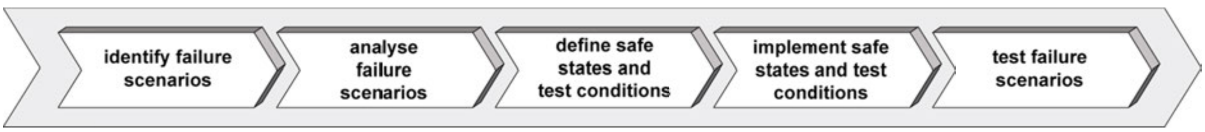

Figure 6. Procedure for failure scenarios for test specifications

The developer has to define safe states and test conditions in step three. Safe states separate between measures to error prevention, error detection and error compensation. Therefore, each failure scenario has to be analysed concerning safe states and test conditions. Due to the fact that in practical applications not all potential failures and errors can be eliminated, failures can be prioritised by compare the additional risk with the effort-benefit ratio. Hence, safe states and test conditions must only be defined for failure over a specified risk priority.
Moreover, the desired safe states and test conditions have to be implemented, e.g. in terms of transitions or monitoring of signals and sensor states. Consequently, the control program ensures the desired behaviour in case of failure. To verify the control program with regard to these scenarios, requirements of the models have to be deduced. Especially, the accessibility of safe states must be ensured by the control program. It must be considered, that failure scenarios require a sufficient

modelling depths. A low level of detail will not suffice to simulate failures, whilst an excessive level of detail takes up to much effort.
Finally, the control program needs to be tested during the virtual commissioning. In an ideal case, the control program counteracts failure whilst switching the system into defined safe states. Otherwise optimisation loops are necessary while the control program ensures desired behaviour.

4.3 Example by the monorail conveying system

In this section, both approaches for application and failure scenarios are exemplified by the monorail conveying system as introduced in section three. In the first instance, application scenarios are illustrated followed by the use of failure scenarios.

Application scenarios

At the beginning of application scenarios representative operation and special operation situations need to be identified. Referring to the demonstrator a guided shuttle movement from the turning to next processing station or loading and unloading the shuttle at the material store are exemplary operation situations. Typical special operation situations of the monorail conveying system are the manual shuttle movement, track or machine maintenance.
Upon detection of operation and special operation situations an analysis of these application scenarios follows to determine functional modules. Possible collaborating functional modules of the above mentioned operation situation (guided shuttle movement) are curves, switches, linear tracks, the infrastructure and the shuttle itself. Then, each functional module need to be analysed in detail. In particular, it is necessary to identify participating functions for the considered application scenario. Concerning the demonstrating application scenario the switch fulfils the function of a dynamic routing by switching between certain track sections, whereas the pressurised air supply is a participating function of the infrastructure module.
Afterwards the characterised operation and special operation situations need to be implemented so that the behavioural models prove the basic feasibility of these situations. For this purpose, a deduction of requirements taking corresponding level of modelling detail into account should occur. Due to the fact that a switch fulfils the function to pilot shuttles between two track sections, it takes two defined positions. These defined positions result in the requirements, which the behavioural models have to meet. For this a low level of detail is sufficient. In contrast picking and placing when unloading and loading the shuttle is not sufficient modelling depth because these operations requires models of used sensor technologies.
Finally, the control program is tested in terms of the application scenarios that are specified with the help of the introduced approach. Depending on the current development status one of the available test methods from model-in-the-loop to rapid control prototyping can be chosen.

Failure scenarios

In analogy to the general procedure as mentioned in the previous section unexpected situations need to be detected. In due consideration of the demonstrating monorail conveying system a derailed shuttle in consequence of an incorrect turned switch or a collision of several shuttles are likely failure scenarios. Wrongly loaded shuttles, incorrect tagged RFID chips or a reduced shuttle speed are possible scenarios as well. The next step in describing failure scenarios is an analysis of identified unexpected situations. The analysis of the example situation revealed that an incorrect turned switch can be the consequence of two basic causes. On the one hand, a technical fault can prohibit a proper functioning of the switch. This includes a lack of pressurised air, a seized switch or a deformed switch, for example. On the other hand, a restrained functionality can be caused by a programming error of the control program. Besides the shuttle, in the considered scenario (derailed shuttle) the linear track, the switch and the infrastructure are involved in the functional modules, as well. Nonetheless, in both cases the control program has to switch to a predefined safe state in case of failure.
Compared to a derailed shuttle the reduced shuttle speed scenario entails minor risks. Therefore, this scenario can be disregarded in the first instance. Subsequently, safe states are exemplarily predefined for the derailed shuttle scenario. Stopping shuttles in the affected track section, closing the section for

any future shuttles or immediately cutting off the power supply are possible measures in case of failure. Measures for error prevention are of even higher importance. Referring to the demonstration scenario an entry into the switch is solely feasible if the switch is correctly positioned. Therefore, the control program needs to monitor relevant sensors or control signals. Some of these specific control parameters are the current switch position, the tracking of moving shuttles or the distance between shuttles.

Both preventive measures and measures for harm-reduction needs to be implemented in a way, that the control program follows predefined instructions as mentioned exemplary above. On the base of specific control parameters transition conditions are determined, e.g. the next shuttle, could only pull into the switch if its position is correct, otherwise the shuttle will be rerouted or finally stopped.

In the end, the developed control program needs to be tested by means of model-in-the-loop, software-in-the-loop, hardware-in-the-loop or rapid control prototyping regarding specified failure scenarios.

5. Conclusion and outlook

In this contribution an approach for scenario-based test specifications for virtual commissioning and the potential benefit has been shown. The test specifications can be used to support the verification of control engineering at an early stage of the development process. In particular, the relevance of two types of scenarios is described in detail. In this context, both the procedure for application scenarios as well as for failure scenarios have been developed. Within the procedural model for virtual commissioning, various simulation methods and approaches for quality management have been briefly discussed.

It has been shown that application scenarios are suitable to verify the control program with regard to desired behaviour in operation situation as well as in special operation situation, e.g. maintenance. Furthermore, this paper illustrated the procedure of deducing failure scenarios for testing and optimising the control program. The test specifications on the basis of application and failure scenarios have been explained by an intelligent monorail conveying system which links several different processing stations and robots. It can be concluded that the test specifications increase the quality of the control engineering at an early stage of the development process. Consequently, the harmonisation of mechanical, electronical, control and software engineering is facilitated. A further aim is a procedure model including synchronisation between the specific development stages and the explicit stages of the virtual commissioning. For example, how the knowledge of the scenario-based test specification influences the detail design stage and vice versa.

Acknowledgements

This contribution was developed in the project “Methodik zur virtuellen Inbetriebnahme von maschinenbaulichen Anlagen auf Basis von objektorientierten Verhaltensmodellen” (Methodology for virtual commissioning of mechanical engineering systems on the basis of object-oriented behavioural models), Univeristy of Paderborn, and was published on its behalf and funded by the DFG Deutsche Forschungsgemeinschaft (German Research Foundation).

References

Brüggemann, B., Bremer, P., "Grundlagen Qualitätsmanagement – Von den Werkzeugen über Methoden und TQM", Springer Vieweg Wiesbaden, 2012.

Deutsches Institut für Normung e.V., "Fehlerbaumanalyse", DIN25424 - Teil 1 -2, Beuth Verlag, 1990.

Drews, G., Hillebrand, N., "Lexikon der Projektmanagement-Methoden", Haufe München, 2007.

Ehrlenspiel, K; "Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit", Carl Hanser Verlag München, 2007.

Heinz Nixdorf Institute (HNI), University of Paderborn, Product Engineering, Fraunhofer-Institut für Produktionstechnologie IPT – Projektgruppe Entwurfstechnik Mechatronik, Unity AG, "Systems Engineering in der industriellen Praxis", Paderborn, 2013.

Hood, C, Wiebel, R., "Optimieren von Requirements Management & Engineering", Springer Verlag Berlin Heidelberg, 2005.

International Council on Systems Engineering (INCOSE), "Systems Engineering Handbook – A Guide for System life cycle processes and acitivites", International Council on Systems Engineering (INCOSE), Version 3.2, 2010.

Isermann, R.; "Fahrdynamik Regelung – Modellbildung, Fahrerassistenzsysteme, Mechatronik", Vieweg und Sohn Verlag Wiesbaden, 2006.

Lochbichler, M., Schmüdderrich, T., Brökelmann, J., Trächtler, A., "Methology for Selecting the Modeling Depth of Object-Oriented Behavioral Models", WASET Engineering and Technology, Zurich, 2012.

Masing, W., "Handbuch der Qualitätssicherung", Carl Hanser Verlag München, 2007.

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., "Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung", Springer Berlin Heidelberg, 2007.

Royce, W., "Managing the Development of large Software Systems", Proc. IEEE WESTCON, Los Angeles, 1970.

Rupp, C, "Requirements-Engineering und -Management", Carl Hanser Verlag München Wien, 2007.

Sauer, O., "Informationstechnik in der Fabrik der Zukunft: Fabrik 4.0 – Aktuelle Rahmenbedingungen, Stand der Technik und Forschungsbedarf", ZWF, No.12, 2011, pp. 955-962.

Schmüdderrich, T., Lochbichler, M., Trächtler, A., "Methodology for selecting the required modeling depth of behavioral models for virtual commissioning", Tagungsband Mechatronik 2013,RWTH Aachen, Aachen, 2013.

Schmüdderrich, T., Trächtler, A., Brökelmann, J., Gausemeier, J., "Procedural Model for the Virtual Commissioning on the Basis of Model-based Design", Proceedings of the 23th CIRP Design Conference, Ruhr-Universität Bochum, Bochum, 2013.

Verein Deutscher Ingenieure (VDI), "VDI-Richtlinie 2206 – Entwicklungsmethodik für mechatronische Systeme", Beuth Verlag Berlin, 2004.

Verein Deutscher Ingenieure (VDI), "VDI-Richtlinie 2221 – Methodik zum Entwicklen und Konstruieren technischer Produkte", Beuth Verlag Berlin, 1993.

Verein Deutscher Werkzeugmaschinenfabriken e. V., "Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen", VDW-Bericht, Werkzeugmaschinenlabor WZL der RWTH Aachen, Aachen, 1997.

Wagner, F., Frey, G., "Hardware-in-the-Loop Simulationen bei kurzfristig zu langsamen Simulationsmodellen", VDI-Bericht, No.1980, 2008.

Wiendahl, H.-P., Reichardt, J., Nyhuis, P., "Handbuch Fabrikplanung: Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten", Carl Hanser Verlag München, 2009.

Wünsch, G., "Methoden für die virtuelle Inebtriebnahme automatisierter Produktionssysteme", Forschungsbericht iwb - Band 215, Technische Universität München, 2008.

Wünsch, G., Zäh, M. F., "A New Method for Fast Plant Start-Up",CARV 2005 1st. International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Utz München, 2005.

Zäh, M, Wünsch, G., Hensel, T., Lindworsky, A., "Feldstudie – Virtuelle Inbetriebnahme", wt Werkstattstechnik online, No.10, 2006, pp. 767-771.

Dipl.-Wirt.-Ing. Marcel Schneider
Heinz Nixdorf Institute, University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
Telephone: +49-5251-606220
Telefax: +49-5251-606268
Email: marcel.schneider@hni.upb.de
URL: https://www.hni.uni-paderborn.de/en/pe/