Dr. Thomas Wagner
Dr. Jan Christoph Wehrstedt
Dr. Ulrich Löwen
Tobias Jäger
Prof. Dr.-Ing. Alexander Fay
Peter Schuller

11

Application and Evaluation in the Automation Domain

The main focus of our work in the automation domain was on the scientific basis and technical implementation of the interaction and the integration in classically heterogeneous systems and development landscapes. This comprised the integration of automation devices and system components, activities, and models of various engineering disciplines, as well as simulation and validation within the scope of a model-based, quality-assured engineering process. The SPES modeling framework formed the foundation for all developments and was extensively evaluated based on multiple real-world case studies in industry. Results show a significant contribution toward the consolidation of domain-specific modeling and systems.

K. Pohl et al. (eds.), *Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology*,
DOI 10.1007/978-3-642-34614-9_11,

11.1 Overview: Application Domain Automation

Purpose and importance of automation engineering

Whether in the manufacturing or processing industry, in mechanical engineering, in transportation, or in logistics — automation engineering plays a key role in controlling and structuring complex systems. Automation engineering describes the design and realization of automated systems for technical processes such as steel production, manufacturing, or refining. It includes process measuring, open and closed loop control, monitoring, alarming, locking, and optimization.

Embedded systems: a core asset in automation

Embedded systems are an essential asset for automating and controlling these processes within industrial solutions and plants. They comprise electrical or electronic devices for measuring and influencing process parameters, hardware components, a communication system, and software for the different automation functions. Embedded systems are used in many different forms reflecting the specific characteristics of processes or industries. With regard to applications of embedded systems in automation, it is important to differentiate between individual devices or single systems, which are developed as mass products, and systems of systems in the form of machines and industrial plants, which are individually implemented in implementation projects for specific customer requirements (Example 11-1). Hence, the same (from an internal point of view) devices or systems deployed in two different systems of systems will appear totally different (from an external point of view) due to the configuration, customization, and integration in applications of different domains with different requirements and constraints.

Example 11-1: *Application areas of embedded systems in automation*

- ❑ **Automation devices**: Generally, standardized, configurable, or freely-programmable hardware components
- ❑ **Machines and apparatuses**: Functional, independent units such as robots, machine tools, or process cells, partly prefabricated and partly fitted
- ❑ **Industrial systems**: Individual configuration of machines, apparatuses, and automation devices for implementing a spatially dispersed process, e.g., power plant, mill, pumping station, factory

Programmable logic controllers (PLC) or comparable specialized microcontrollers are deployed as hardware. These are electronic control systems used to automate a variety of technical processes featuring multiple input and output arrangements, real-time signal processing, deterministic logic execution, modular hardware configurability, minimized space, and extended temperature ranges. PLCs are programmed via an interconnected computer using domain-specific standardized programming languages.

Integration of embedded systems is the key challenge in automation

Automation engineering is the key for being able to manufacture top-quality and affordable products in high-wage countries. However, embedded systems in automation engineering solutions require the adaption and integration of intricately connected mechanical, electrical, and IT units from functional, logical, technical, and spatial aspects (see Example 11-2). Conversely, there are stringent requirements for the ability of the individual automation devices and systems to be combined and integrated.

Example 11-2: *Characteristics of embedded systems in automation*

- ❑ Specific combination of software and system components from various manufacturers that are partially proprietary and partially comprised of old and new components
- ❑ Collaboration and integration of the models of many disciplines
- ❑ Integration and maintenance of existing software and system components with new systems and technology changes in the life cycle

Integration must be driven by models

The control of this integration complexity requires effective collaboration across the domain. In this case, the software takes an increasingly important role because it forms the last link during the implementation or initial operation of the entire system and thus, must ensure the proper interaction of all disciplines. Moreover, the automation system can only be fully tested in conjunction with the technical process, i.e., following its implementation on-site. An early integration, therefore, can only occur on the model level. However, currently, the individual models of the disciplines are still being developed largely in isolation. These models must be interlinked, whereby it is necessary to ensure consistency throughout the individual disciplines, the entire life cycle of the automation solution, as well as throughout the various levels of automation.

Model-based development is not only expected to improve efficiency in the interdisciplinary development process of automation engineering through networking and parallelization; in addition, appropriately connected models can be used tactically to make requirements and design decisions transparent across all disciplines at an early stage. They can also be used to secure performance parameters of the automation solution early in the proceedings and thus, to minimize development risks and costs [Wagner and Löwen 2010].

11.2 Evaluation Strategy for the Domain

Ongoing coordination of modeling

From the perspective of automation engineering, a promising approach was to start comparing the terms and concepts already proven in the domain with the SPES modeling framework in order to lay a scientific foundation. Therefore, the characteristics and cross-references of discipline-specific models were analyzed at the beginning of the project through the analysis of case studies and actual automation projects of industry partners. The requirements of the domain for an integrated, model-based development process were also determined. In this context, particular focus was placed on the alignment of the various terminologies and model concepts from software and automation engineering, and thus the foundation for the transferability of requirements and solutions was created.

Development-accompanying evaluation of the metamodel in the domain

The concepts of the SPES modeling framework were compared to existing industrial approaches and optimized parallel to the development process. For the evaluation, automation system models were assigned to viewpoints of the SPES modeling framework and integrated in automation engineering development processes. The concepts and methods of the SPES modeling framework were represented through prototypical modeling of selected case studies and tools with the participation of domain experts and the academic partners.

11.3 Overview of Activities and Results

In this section, a general overview of the evaluation context and the conducted evaluation activities is given.

Tab. 11-1 *Industries of the automation domain and SPES case studies*

Area	Explanation	SPES Case Studies
Process plants	Automation of physical or chemical transformations of substances, materials, or energy by a sequence of continuous flow processes	Pumping station plant for water distribution (Siemens AG)
		Hot rolling steel mill (Siemens AG)
Manufacturing plants	Automation of procedures acting upon the forming, working, and joining of materials or items by a disconnected transition of discrete process operations	Test bench for wings (E4You)
		Train control system (Siemens AG)
		Cylinder production (Siemens AG)

11.3.1 Overall Context of the Activities

Best practice modeling concepts for all industries of the automation domain

The activities focused on the development and transfer of best practice methods between the modeling of automation systems and the SPES modeling framework. Multiple complementary case studies from various application areas of the domain (Tab. 11-1) were used to evaluate the SPES modeling framework. In addition, domain-specific characteristics and enhancements of the metamodels were developed and implemented.

11.3.2 Model-Driven Development of Automation Devices

The focus was on the development of best practices and guidelines for the systematic introduction of model-based processes [Fieber et al. 2009]. Moreover, methods for assuring the quality of models in the industrial environment were developed, and the software development for automation devices was observed in particular in these methods [Arendt et al. 2011]. The SPES modeling framework was tested for its applicability on the basis of the case study *Train control system*. In order to deliver proof of the usability of the results, a prototype for the automated quality assurance of models was implemented for the case study and has already been successfully applied in further projects of Siemens AG [Arendt and Taentzer 2012].

11.3.3 Model-Driven Integration and Simulation of Embedded Systems in Industrial Systems

Here, the focus was on the application of the SPES modeling framework for the interdisciplinary modeling (engineering, electrical engineering, software) of industry systems with regard to the integration and dependencies between the different disciplines involved in automation engineering [Jäger et al. 2011], and with regard to the use of models for the improvement of the interdisciplinary collaboration (cf. [Wagner et al. 2011, Fay et al. 2011]). An additional emphasis was on the validation of the embedded systems in systems through the use of functional, logical, and technical models of the systems for simulations [Wehrstedt et al. 2011].

The application of the SPES modeling framework to the description methods of automation engineering and its applicability were evaluated on the basis of three case studies: *Hot rolling steel mill*, *Pumping station*, and *Cylinder production*. Aside from expert analysis, the prototypical re-modeling of actual systems was applied in the studies for evaluation [Lüder et al. 2010, Wagner et al. 2011]. The activities were accompanied by practical demonstrations. The results of the evaluation are described in further detail in Section 11.4.

11.3.4 Data Models for Automation Runtime Platforms

At the center of our work were the design and the implementation of a runtime platform to which various automation devices and development tools could be connected. Therefore, PLCs (e.g., radCASE), process visualizations (e.g., embedded graphic XiBase9), and test environments (e.g., YAVE) were integrated using the Embedded4You middleware GAMMA on the basis of a central modeling approach.

This data model represents the SPES modeling framework completely and abstracts the system components of embedded platforms, such as hardware, I/O structures, operating systems, and communication. It describes the transition from platform-independent (PIM) to platform-specific data models (PSM). A PIM describes the model of the logical viewpoint in the SPES modeling framework, whereas a PSM is a model of the technical viewpoint of the SPES modeling framework. These models are connected via a mapping relationship. The models of the various abstraction layers and viewpoints require execution semantics to be able to validate properties of the system in early stages of development. Therefore, the modeling approach also contains platform-specific artifacts for the execution, such as process and I/O variables, temporal processes, or memory dependencies. Together with the test environment, the data-centric model enables simulation models to be tested with MATLAB/Simulink and software or hardware in the loop.

The results are applied in diverse embedded platforms and test systems. Special platforms were implemented to prove the modeling on typical hardware environments of automation. The evaluation occurred in the *High-lift test bench* case study (modeling of the high-lift test of wings). In that context, a flexible microTCA hardware platform was built. The functional scope of the case study is scalable to modeling in automation.

The proof of concept was shown by implementing and testing the runtime platform for the case study *High-lift test bench*. The technical implementation of the model only occurred in the form of an example within the scope of the project due to high costs. As far as the development environment is concerned, the most significant concepts were developed and implemented using prototypes. The activities were accompanied by scientific work for modeling temporal aspects and the introduction of the results into the standard [VDI/VDE 2657].

11.4 Application and Evaluation of the SPES Modeling Framework

Collaboration of scientific and industrial partners

The models and results presented in the following consolidate the work from numerous case studies that were jointly developed by Siemens AG, the Helmut Schmidt University/University of the Federal Armed Forces Hamburg, OFFIS e.V., the University of Duisburg-Essen, and the Technische Universität München. For the purposes of clarity and confidentiality of real project information, they will be explained based on the simplified demonstration model of a production technology system.

11.4.1 Domain-Specific Challenges

Software within automation systems is becoming more complex and crucial to such systems [Achatz and Löwen 2005], resulting in some important domain-specific challenges — this was the focus of the evaluation.

Integration of multiple engineering disciplines

A variety of engineering disciplines are involved in the realization of an automation system: electronics, mechanics, process technology, embedded hardware and software. The disciplines all require their own models and methods that are specialized for their particular design purpose. However, design decisions and results of different disciplines depend on each other. A modeling approach has to offer system models that span disciplines, as well as views onto these models that support discipline-specific methods and models and ensure the consistency and traceability of the data of the different views.

Scalability for large number of entities

Complexity also arises from the great number of functions, signals, and devices in automation systems. Models, and especially tools, have to stay in control and must support the engineers to stay on top of things.

Process integration of modeling approach

Due to their complexity, in practice, automation projects are process-driven and consist of different phases in which the projected system is designed, refined, and made more specific step-by-step. Important phases are concept engineering (for bid preparation), basic engineering, detailed engineering, and installation and commissioning. In each phase, different objectives and requirements have to be supported by models (see Tab. 12-2). For example, a project must be able to estimate the major cost items as early as possible for the purposes of bidding (e.g., equipment, engineering and construction). A modeling approach must allow this information to be revealed without excessive effort.

Tab. 11-2 *Typical phases of an automation engineering project*

Phase	**Concept Engineering**	**Basic Engineering**	**Detailed Engineering**	**Installation & Commissioning**
Aim	Plant is feasible	Plant to be contracted	Plant can be built	Production can start
Typical Tasks	1. Clarify technical scope 2. Analyze requirements & risks 3. Design solution concept 4. Assess effort & quantities	1. Design technical process, plant architecture, and construction 2. Initial technical specifications & calculations 3. Simulation and validation	1. Design all systems 2. Fully specify equipment 3. Purchase, manufacture, and implement systems and software	1. Assemble mechanical, electronic, and automation parts 2. Deploy software 3. Check, inspect, and test every operational component

11.4.2 Introduction to the Case Studies

Case study: Cylinder machining unit

The demonstrator represents a processing cell in a production system for manufacturing cylinder heads and offers the possibility of mapping and simulating planning processes and models of actual industrial systems. The reduced complexity of the demonstrator allows representative results to be achieved in a short time, and these results are then transferable to the actual planning process for industrial systems.

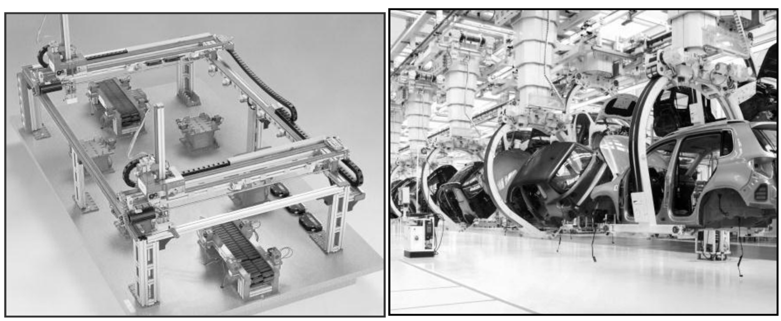

Fig. 11-1 *Demonstration model and example of a production system*

The demonstrator consists of four stations for work piece processing: two conveyor belts and two gantry cranes for transporting work pieces. Automation enables multiple work pieces in the cell to be processed concurrently. To achieve this, optical sensors for determining the work piece position are installed, as well as push buttons for determining the crane position. For the automation, a PLC S7-400 is used, and is connected via PROFIBUS to a total of 52 sensors (position measuring, work piece identification) and actuators (drives).

The system was engineered using plant engineering tools COMOS, Mechatronics Concept Designer, and the automation system SIMATIC PCS7. In addition to the configuration of the hardware and communication, this engineering also included the creation of a control program and the configuration of the operator control panel.

11.4.3 Mapping of the SPES Modeling Framework to the Concept and Description Methods for Automation

4-tier metamodel classifies all entities in automation modeling

To make the SPES modeling framework compatible for modeling automation engineering, the common entity classes, views, relationships, and concepts of the domain-specific models have to be assigned to those of the SPES modeling framework. To enable this assignment independent of the various discipline-specific and industry-specific modeling languages, a generic metamodel of automation engineering was developed (see Fig. 11-2) [Strube et al. 2010]. It describes all elements to be modeled and their relationships relevant from an automation engineering perspective for the functionality and integration of a system. The 4-tier model can be used to interlink the SPES modeling framework to the domain-specific modeling languages. In this context, tiers 3 and 4 represent the automation system. The modeling of the system environment on tiers 1 and 2 is of particular importance, not only for compiling the requirements from the engineering process and resources to the automation, but also for validating all requirements. This requires an overall analysis of the system considering the interaction of processes, resources, hardware, and software.

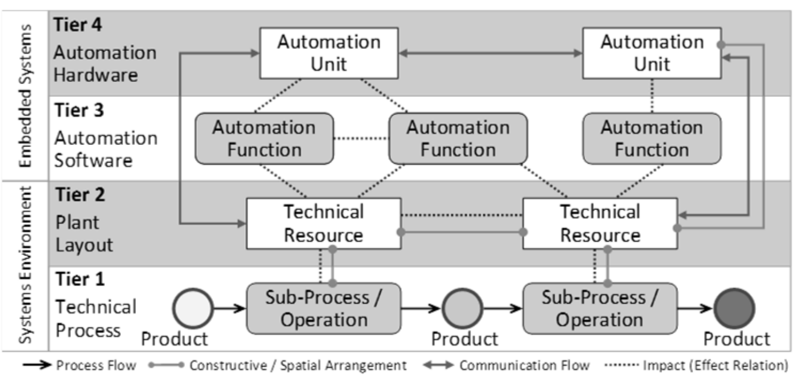

Fig. 11-2 *4-tier metamodel for automated plants [Strube et al. 2010]*

Metalevel domain compatibility: SPES viewpoints cover all entity classes of automation models

Tab. 11-3 shows the assignment of the entity classes of automation models to the viewpoints of the SPES modeling framework as well as the associated domain-specific modeling languages. There is no one to one assignment of the domain-specific modeling languages to the viewpoints of the SPES modeling framework. Many automation engineering models contain functional as well as logical and technical aspects, e.g., the system layout or the process and instrumentation diagram. The requirements viewpoint is not listed because it is still minimally established in the domain and the SPES concepts can be used without further adaptation.

Tab. 11-3 *Mapping of SPES viewpoints to automation entity classes*

Viewpoint	Automation Entity Class		Domain-Specific Models
Functional Viewpoint	Tier 1: Technical process	Sequence of process operations with which raw materials are transformed into final products	Basic/process flow diagram Bill of operations (BOO) Functional architecture
Logical Viewpoint	Tier 2: Plant layout	All types of technical resources (machines, apparatuses) for executing the technical process	Plant breakdown structure Plant layout (CAD model) Material flow diagram Process & instrumentation diagr.
	Tier 3: Automation software	Necessary automation functionality for controlling the process of tier 1 with the resources of tier 2	Process control hierarchy Measuring/set point list State chart/Petri net/SysML Function charts/blocks
Technical Viewpoint	Tier 4: Automation hardware	All devices for controlling and monitoring of the technical process by executing automation software	Parts list: drives, instruments Hardware/network configuration Input/output signal list Wiring & mounting diagrams

Models in automation have a strong physical connection

Only a limited number of technological processes are available for particular process operations, which in turn affect the type and features of technical resources and automation functionality. From the bill of operations of the cylinder processing cell presented in Fig. 11-3, it is obvious that the individual process steps can only be executed by certain types of machines. Further technical requirements arise through the given process templates and the parameters of process operations (e.g., temperature, forces, velocity, geometry). For example, in the case of simultaneous movement of the gantry cranes in the cylinder machining unit, collisions must be avoided. These types of system requirements are modeled according to the SPES modeling framework as contracts (Fig. 11-3), and can therefore be taken into consideration in the development process so that future integration or functional problems can be minimized.

The logical viewpoint serves as a platform-independent description of the logical system architecture. In automation engineering, this can be mapped to the design of the system layout and the necessary automation functionality according to the process flow (see Tab. 11-3).

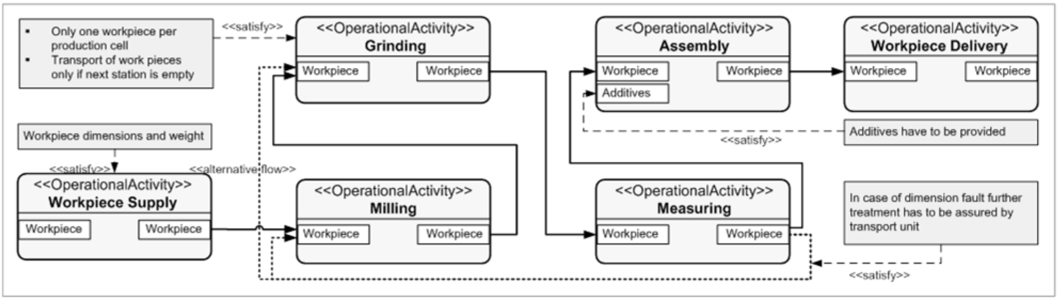

Fig. 11-3 *Functional view: Bill of operations of the cylinder machining unit*

Logical viewpoint models shaped by technical platform

Evaluation showed that, in contrast to the SPES modeling framework approach of a platform-independent logical system model, the layout of automation systems (tier 2) must consider the technological platform to some extent. For example, the type of technical resources needed according to the selected technological process for a process operation must be determined, which in turn also significantly affects the type and features of the automation functionality (tier 3). For example, based on the decision that gantry cranes have to be used to transport work pieces in the system layout of the processing cell instead of robot arms (Fig. 11-4), the number and type of sensors and actuators, and thus, the process signals as well as the type of automation software (e.g., the logical process control instead of numerical movement control) are determined. The automation functionality cannot be modeled independently of the platform because it is dependent on the features of the hardware (temporal relationship, bus systems) and the sensor/actuator interfaces. For the logical viewpoint, therefore, few independent models currently exist in automation engineering; rather the elements of the logical view are primarily described in technology-oriented models.

Due to the fact that the technological process forms the common basis for all disciplines involved in the planning process and their models, the dependencies between the process, the technical resources, and the automation system must also emerge from the functional and logical system model. On one hand, this approach increases the degree of the overall description of the system and in the process, also helps to avoid misunderstandings and integration issues between the disciplines. The challenge from the perspective of automation engineering was in finding suitable modeling strategies so that, on one hand traceability beyond viewpoints exists, and on the other, a viewpoint is closed. For this reason, the 4-tier metamodel also contains a classification of technically relevant relationships between the entities (Hint 11-1). Furthermore, a method for analyzing and modeling technical dependencies when engineering a system was developed [Jäger et al. 2011, Strube et al. 2011].

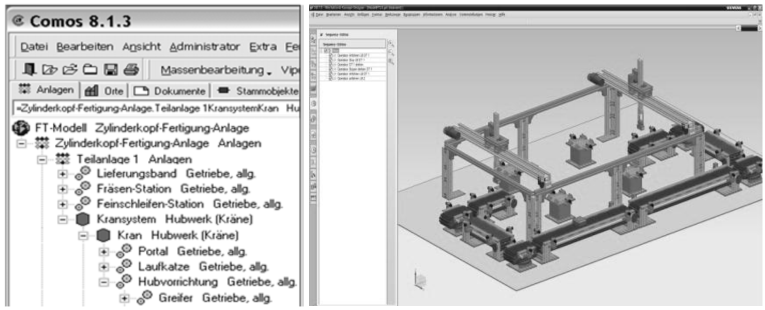

Fig. 11-4 *Logical view: plant breakdown structure modeled in COMOS (left) and 3D layout in Mechatronics Concept Designer (right)*

Functional and logical design is dependent on the technical aspects

Hint 11-1: *Classification of model relations of automation entities*

- **Process flow** represents the transport of items, substances, or energy between process operations and/or storage
- **Structural and spatial associations** represent the spatial or constructive assembly of entities in the plant (e.g., shaft, screw) or mechanical connections for transporting material or substance (e.g., pipeline, cable)
- **Communication relations** between units represent the exchange of signals/information (test value, set point, function call, etc.), e.g., wire connection between sensors and actuators of a technical resource and an automation device.
- **Relationships of effect** represent functional dependencies between technical resources as well as between automation functionality with the controlled resource and the process operation executed

Structural and communication relations are modeled using the SPES modeling framework concepts *logical component, port, connection, mapping,* and *realization*. To model simple flow and relationships of effect, the concepts *allocation, mapping,* and *contracts* are applied. Fig. 11-5 shows this in the example of the logical model of the automation functionalities of the cylinder processing station. The automation functionalities on the next abstraction layer are hardened through the domain-specific modeling languages *continuous* or *sequential function charts,* and thus the transition for the implementation of the logical components is created with logical PLC programming languages (function blocks), as shown in Fig. 11-6.

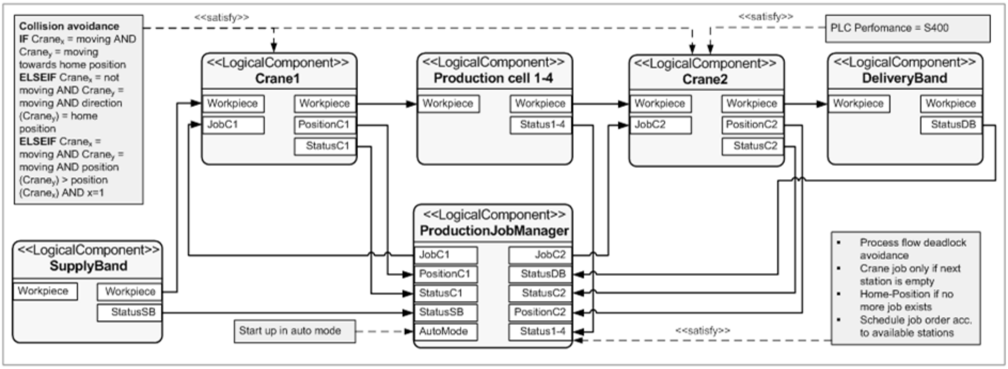

Fig. 11-5 *Logical view of a high abstraction layer: functions, interaction, and contracts*

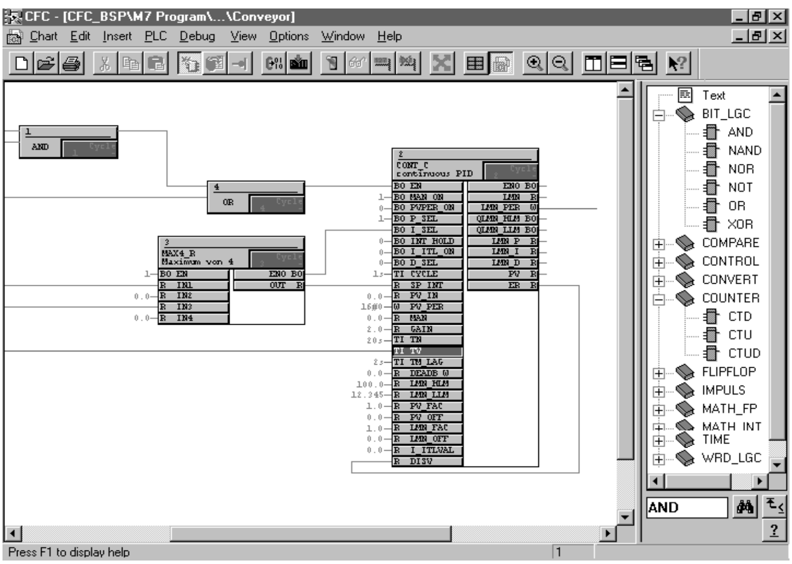

Fig. 11-6 *Logical view of a lower abstraction layer: crane control: continuous function chart*

Modeling of relations of automation entities

Within the technical viewpoint, the logical components of the logical model are allocated to equipment and the software is distributed to the automation devices. The main entities are technical components such as electronic control units, for example, PLC, control cabinet, operator station, and systems, as well as actuators and sensors (Fig. 11-7). This network represents the hardware platform to which the logical components must be allocated (Fig. 11-8). Automation hardware is typically selected from catalogs and the software is implemented with appropriate tools specific to the hardware. The specific machines and apparatuses are selected for the technical resources. They include additional technical components for discipline-specific subsystems such as power supply, hydraulics, etc.

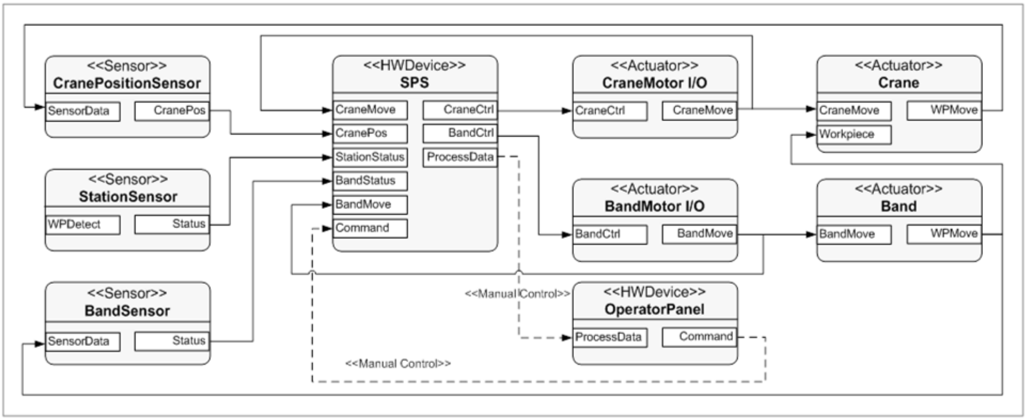

Fig. 11-7 *Technical view: hardware layout*

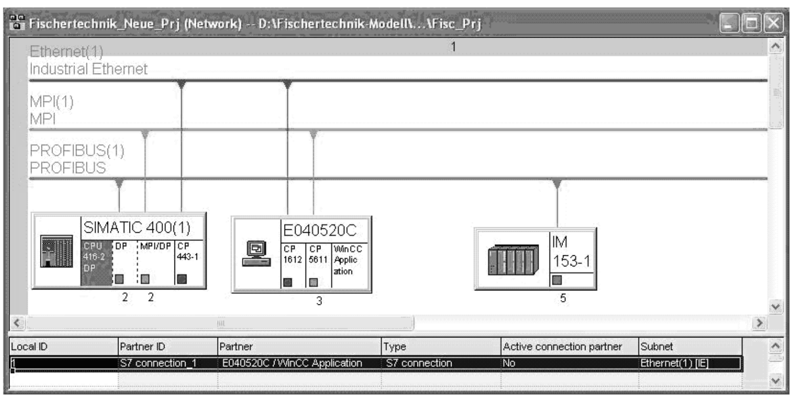

Fig. 11-8 *Technical view: network layout in SIMATIC NetPro*

Through the formalization of the process conditions and relationships of effect, both can be analyzed automatically. In this way, it is possible to determine, for example, whether logical components that depend on each other are also compatible with each other. In order to enable a complete review of the functionality and of the embedded system during technical process control prior to the actual systems deployment, solution concepts have been developed. For example, executable simulation models for process and automation can be generated from the interdisciplinary complete model [Wehrstedt et al. 2011]. These are used to validate the functional and logical models of the system with the use of simulation tools — both in an early phase as well as during the virtual initial operation. For example, it is necessary for the cylinder machining unit to test whether the control operates without hindrance or collision.

11.4.4 Evaluation of the Abstract Modeling Concepts

In addition to the application of case studies, the modeling concepts of the SPES modeling framework were compared to those of common design methods, modeling languages, and tools of automation. The comparison was based on a criteria catalog with defined and weighted instances (including examples) of conceptual abstractions. The validity and completeness of the criteria catalog for analysis and comparison of modeling and tooling concepts has been proven in more than 25 evaluations for several Siemens business units in recent years.

An overview of the results is presented in Tab. 11-4. In summary, a complete consideration and specification of the concepts *views, hierarchy,* and *abstraction* can be found in the SPES modeling framework. It also became evident that the modeling concepts *modularization, viewpoints, aspects,* and *dependency/mapping* were not yet completely specified in both SPES and automation engineering. The concepts *reuse* and *mechatronics* are only applied in the automation domain and are not currently considered in the SPES modeling framework.

11.4.5 Methodological Approach for the Engineering of Systems on the Basis of the SPES Modeling Framework

The SPES modeling framework is an architecture model and does not provide a process model. Due to the fact that automation projects are strongly driven by processes, a suitable approach for the engineering of automation systems based on the SPES modeling framework was derived. In the SPES modeling framework, the engineering phases *concept, basis,* and *detailed engineering* are supported by the viewpoints. The installation and commissioning phase of standard process models is currently not provided. The models to be created in this phase, however, do not require currently unknown modeling concepts and it was therefore possible to depict them in the SPES modeling framework (Fig. 11-9 left).

Tab. 11-4 *Evaluation of abstract modeling concepts*

Concept	Emphasis		Explanations
	SPES	AUT	
Hierarchy	Very High	Very High	Hierarchization through composition and decomposition is used in both domains. There is a common understanding in the areas of consideration and specification.
Abstrac-tion	High	High	Is substantial in the SPES modeling framework, although the type of abstraction is not specified further. For example, the role concept significant for the automation can be depicted with a functional viewpoint and implementation relationship, though an explicit type of artifact is missing: *role* or *abstract component*.
Modulari-zation	Med.	High	Modularization has to be applied in the SPES modeling framework when hierarchizing. However, the consideration and specification of a hierarchy-wide modularization, such as is implemented in the automation domain with the aid of the group concept (logical areas), is lacking. This would correlate to a viewpoints and cross-aspect viewing aggregation.
Discipline-specific views	Med.	Med.	The viewing concept is implemented in the SPES modeling framework entirely. The discipline-specific design, however, is not sufficiently supported. Aspects allow the internal classification of information within a component, though not an explicit component-wide classification or summary of individual aspects, interfaces, and connections.
Depen-dency	Low	Low	The mapping concept with the characteristics *implementation, allocation,* and *link to external data* is currently not fully specified. For example, extensions for the representation of complex dependencies with class, features, and functionality are required.
Reuse	None	High	The SPES modeling framework does not currently provide a reuse concept. Libraries, instance, and inheritance are required to support the reuse of engineering artifacts.
Mecha-tronics	None	Med.	Artifacts are classified in the SPES modeling framework. However, in automation, an artifact frequently comprises information from multiple viewpoints that must be considered cohesively because a significant added value of the modeling arises.

In automation engineering, firstly the system requirements are analyzed (decomposition). As explained by means of the case study, requirements are not only based on functions of the functional viewpoint, but also on elements of additional viewpoints. That means that a sequential arrangement of the design within a viewpoint or a level is not sufficient. The use of a parallel process instead allows temporal overlapping of the design activities of the different disciplines involved. In addition, it must be possible to integrate solution elements while increasing the level of abstraction (bottom-up). This results in the process flow as shown in Fig. 11-9 (right).

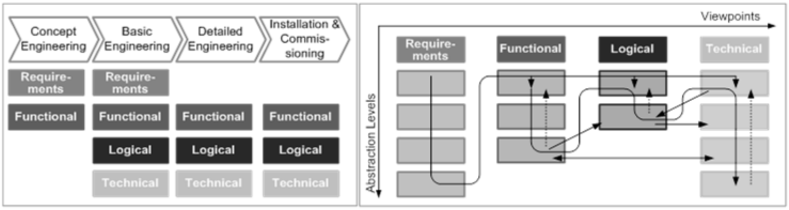

Fig. 11-9 *Mapping of the SPES modeling framework to the engineering process*

11.5 Summary

The goals of the automation domain were to optimize, interlink, and increasingly automate the engineering of automation systems using suitable models based on the SPES modeling framework. This requires integration of the discipline-specific models within the scope of structured, collaborative, and scalable engineering processes, also considering the physical and technical constraints of the automated system.

The concepts of the SPES modeling framework were mapped to the specific modeling languages of the automation domain. In this process, through intensive coordination with representatives from the automation domains and academic partners within SPES, it was found that the terms of automation engineering and those of the SPES methodology correspond well. Based on results from applying the SPES modeling framework in the context of several domain-specific case studies, we conclude that the automation engineering 4-tier metamodel of automation can be represented within the viewpoints and abstraction layers of the SPES modeling framework. Based on results from case studies, we were able to demonstrate that the SPES modeling framework can be applied in the automation domain.

Results also showed that the SPES modeling framework strengthens abstraction, hierarchy, and separation of concerns in interdisciplinary modeling and thus promotes interdisciplinary collaboration, which can be underlined from experiences from recent projects. Moreover, the emphasis on the functional and logical views supports automation engineering for shifting efforts from implementation into the early project phases, helping to reduce project risks as well as costs for the correction of errors and design decisions [Wagner et al. 2011]. However, there is currently no guideline for the application of the SPES modeling framework in the automation engineering process. Therefore, the SPES modeling framework has to be adapted further with regard to its applicability in the automation domain. Specifically, this could be achieved through the integration of upcoming domain-specific challenges such as mechatronic design and design-by-reuse and by support through domain-specific application guidelines.

11.6 References

[Achatz and Löwen 2005] R. Achatz, U. Löwen: Industrieautomation. In: P. Liggesmeyer, D. Rombach (Eds): Software Engineering eingebetteter Systeme: Grundlagen, Methodik, Anwendungen. Munich: Elsevier, Spektrum Akademischer Verlag, 2005, pp. 497–525.

[Arendt and Taentzer 2012] T. Arendt, G. Taentzer: Model-driven engineering of composite model refactorings. In: Proceedings of the 34th International Conference on Software Engineering, 2012.

[Arendt et al. 2011] T. Arendt, S. Kranz, F. Mantz, N. Regnat, G. Taentzer: Towards syntactical model quality assurance in industrial software development: Process definition and tool support. In: R. Reussner, M. Grund, A. Oberweis, W. Tichy (Eds.): Software Engineering 2011: Fachtagung des GI-Fachbereichs Softwaretechnik, February 25-25, 2011 in Karlsruhe. Köllen Druck+Verlag, Bonn, 2011.

[Fay et al. 2011] A. Fay, T. Jäger, T. Wagner: Systematische Erfassung und Bewertung von gewerkeübergreifenden Schnittstellen in Engineering-Workflows. In: Proceedings des 8. Symposiums Informationstechnologien für Entwicklung und Produktion in der Verfahrenstechnik, Frankfurt am Main, Germany, 2011.

[Fieber et al. 2009] F. Fieber, N. Regnat, B. Rumpe: Assessing usability of model driven development in industrial projects. In: T. Bailey, R. Vogel, J. Mansell (Eds.): 4th European Workshop on "From code centric to model centric software engineering: Practices, Implications and ROI" (C2M). University of Twente, Enschede, 2009.

[Jäger et al. 2011] T. Jäger, A. Fay, T. Wagner, U. Löwen: Mining technical dependencies throughout engineering process knowledge. In: Proceedings of the 16th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'2011, September 5-9, 2011, pp. 1-7.

[Lüder et al. 2010] A. Lüder, L. Hundt, M. Foehr, T. Wagner, J. Zaddach: Manufacturing system engineering with mechatronical units. In: Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA'2010 September 13-16, 2010, pp. 1-8.

[Strube et al. 2010] M. Strube, A. Fay, S. Truchat, H. Figalist: Funktionale Anlagenbeschreibung als Basis der Modernisierungsplanung. In: VDI-Bericht Nr. 2092 Tagungsband VDI-Kongress AUTOMATION 2010, Baden-Baden, Germany, 2010.

[Strube et al. 2011] M. Strube, A. Fay, S. Truchat, H. Figalist: Durchgängige Modellunterstützung bei der Modernisierung komplexer Automatisierungssysteme. In: VDI-Bericht 2143 Tagungsband Automation 2011, Baden-Baden, Germany, 2011.

[VDI/VDE 2657] Verein Deutscher Ingenieure: VDI/VDE Guideline 2657 - Middleware in industrial automation (draft for guideline), ICS 35.240.50, 2010.

[Wagner and Löwen 2010] T. Wagner, U. Löwen. Modellierung: Grundlage für integriertes Engineering. In: VDI-Berichte Nr. 2092 Tagungsband VDI-Kongress AUTOMATION 2010, Baden-Baden, Germany, 2010.

[Wagner et al. 2011] T. Wagner, T. Jäger, A. Fay, H. Figalist: Systematische Risikominimierung im Engineering mit Abhängigkeitsanalyse und Schlüsseldokumenten – Vorgehen und Ergebnisse einer Fallstudie zur Erfassung der gewerkeübergreifenden Informationsschnittmenge im Engineering automatisierter Anlagen. In: VDI-Bericht 2143 Tagungsband Automation 2011, Baden-Baden, Germany, 2011.

[Wehrstedt et al. 2011]J.C. Wehrstedt, R. Rosen, A. Pirsing, C. Dietz: Simulation Based Engineering – Frühzeitige Validierung von Anlagenkonzepten. In: J. Gausemeier, F. Rammig, W. Schäfer, A. Trächtler (ed.): 8. Paderborner Workshop Entwurf technischer Systeme. HNI-Verlagsschriftenreihe Band 294, Paderborn, Germany, 2011.