

Engineering of Automation Systems using a Metamodel implemented in AutomationML

Benjamin Brandenbourger, Milan Vathoopan, and Alois Zoitl
fortiss GmbH

Guerickestr. 25, 80805 Munich, Germany
{brandenbourger, vathoopan, zoitl}@fortiss.org

Abstract—The commission and adaptation of modern automation systems is cost-intensive, error-prone and requires expertise. A large part is attributed to the engineering and programming process of manufacturer-specific components. Unifying the way of describing and controlling hardware results in a higher flexibility of the utilized hardware. This paper introduces a metamodel which allows a more efficient and flexible engineering of industrial automation systems throughout the system life cycle. The resulting model-based engineering approach utilizes highly abstracted models of automation components stored in the metamodel to form more specific models. By applying inheritance and classes, paradigms of object-oriented programming are followed to ensure better re-usability and interchangeability of modeled components. For evaluating the proposed concept the corresponding engineering data model is implemented in AutomationML.

I. INTRODUCTION

Manufacturing industries are going through drastic changes these days. With increasing customer requirements [1] and market saturation [2] companies are forced to provide more value-adding services with their products. Due to the dynamic and variant nature of customer requirements, the products are getting more customized and personalized while their life cycles are getting shorter and shorter [3]. In parallel, labor cost [4] is also increasing day by day.

A manufacturing company's key to success in this modern scenario is its ability to design, test, and market high quality and low cost products in a short time frame [3]. This in effect obligates the plants to be equipped with flexible automation systems [5] with high degree of adaptability [4] and upgrades over time. Adaptions to an automation system involve different disciplines and need to be performed in a minimum time frame to keep the downtime as short as possible. So it is crucial that these automation systems facilitate inter disciplinary interaction [6] and integration of the disciplines involved in their development and maintenance throughout their lifetime. In the software engineering domain, model-based engineering is proven to be efficient in data exchange, reuse of models, and the development process as a whole [7]. However, the concept is still at an initial stage of development in the industrial automation domain. When considering a model-based development in the industrial automation domain, the model itself is found to be complex with information from multiple engineering domains. It has to be compact and capable of incorporating semantic relations among the disciplines

involved. So more effort is needed in developing such a model [8]. The model, once established, has to be consistent throughout the life cycle of a plant and should enable data transfer and collaboration among different engineering groups associated with the process, which makes maintaining such a model also difficult.

This paper introduces a model-based engineering approach, encompassing the whole engineering process of an automated manufacturing plant. A standardization committee is introduced, which is responsible for establishing and maintaining a standardized and consistent engineering data model for an automated manufacturing plant. This standardized engineering data model acts as a metamodel for development, operation and maintenance of automated manufacturing plants. Furthermore a tripartite standardized engineering process using this metamodel is put forward. For evaluating the proposed concept, the corresponding engineering data model is implemented in the automation data exchange standard AutomationML (AML) [9]. Moreover, the interdisciplinary interaction and integration of the involved disciplines are analyzed using an application example and the overall improvement in the engineering process is examined.

The rest of the paper is organized as follows. Section II gives an overview of available work in the field of model-based engineering in the industrial automation domain. Section III introduces our engineering approach and the proposed metamodel. How to apply the metamodel is explained in Section IV whereas Section V describes its implementation with AutomationML. Section VI evaluates the AutomationML-based model for data transfer and interdisciplinary interaction. Section VII concludes the paper giving overall results and future scenarios.

II. RELATED WORK

Requirements of an advanced engineering process in the industrial automation domain has been in literature for a quite long time. Vogel-Heuser et al. [7] identify modularity, reuse, and variant management as the important requirements from industry's point of view. Feldman et al. [10] further develop this idea to increase the efficiency and reuse of both hardware and software components. A modular engineering approach is proposed considering the variant management and increase of reuse in the manufacturing domain. A module is an aggregation of components from different disciplines which are involved in the process and thus enables collaboration

of the involved disciplines. The concept of our engineering approach based on a novel metamodel is derived from the aforementioned works by enforcing modularity and reuse.

Eppinger et al. [11] introduce a strategy for developing an engineering data model by analyzing the relationship between the many design tasks to be performed and then reorganizing them. Ovtcharova et al. [12] suggest the importance of developing a functional model. Since product function remains the same for all, they propose a functional model-based ontology as a solution to the significant gaps existing in cross domain engineering and to improve the collaboration process. Another approach from Hackenberg et al. [8] investigates an engineering data model that fits into the conception and refinement phase. They introduce a model-based methodology that enables data exchange, simulation support, and also behavior modeling for synchronization and collaboration between these phases. The german VDI guideline 3695 [13] presents methodologies for engineering organization improvements. Different advices concerning messages, flow charts, operation procedures, and devices are discussed in here. However these approaches mainly concentrate on the model itself or its possible application methodology. A real application scenario considering the model is not in the scope of these works.

AML has been explored by many people for implementing models and their applications in the industrial automation domain. Himmler [6] proposes a functional model-based engineering approach consisting of three steps termed define, standardize and realize. In this approach, standardized engineering interfaces are developed using AML. However, only the functional aspects which evolve from the initial stage of plant engineering are considered in his work and a complete cross domain view is not presented. Nevertheless, the validation of the proposed functional model-based approach proves that this approach has positive effects on the process cycle times, data quality and data redundancy. Berardinelli et al. [14] introduce a formal engineering data model and validate the co-evolution of the model using AML, but a complete cross domain engineering data model with AML has not been evaluated so far.

III. INTRODUCTION OF THE CONCEPT

In this section, we formulate the requirements for the efficient engineering of manufacturing plants and derive an envisioned engineering process. A metamodel supporting this envisioned engineering is presented in the last subsection.

A. Requirements for an Efficient Engineering Process

Nowadays, the engineering of automation systems with models of components used in the plant requires several crucial features to increase efficiency and to reduce development costs. Jazdi et al. [15] state that re-usability of modeled components and efficiency during engineering can be increased by separating project-specific engineering tasks tailored to one specific plant from project-unrelated engineering tasks (R1) which can be reused in different projects. Continuous data management (R2) along the different stages of the engineering

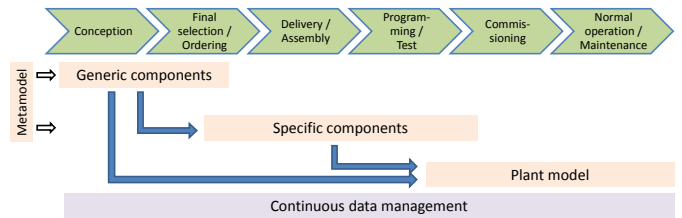


Fig. 1. Engineering phases of an automation system and envisioned engineering

process [16], as well as supporting cross-domain modeling [17] (R3) improves the efficiency of engineering in two ways: first, by avoiding inconsistencies (R3.1) in the modeled components and also the modeled automation system [18] and second, by minimizing redundant engineering tasks (R3.2) in different engineering tools. Therefore, a standardization (R4) on multiple levels such as mechanical, electrical, and software needs to be followed. Feldmann et al. [10] state that software tool support (R5) is necessary to help engineers develop their solution. The tools should assist in identifying appropriate modules (R5.1), combining variable modules (R5.2) presented in [19], and executing simulations (R5.3). By meeting these requirements, the (economic) efficiency and planning reliability can be improved.

B. Envisioned Engineering Approach

Based on the requirements, an envisioned engineering approach is presented in this section. The engineering process of automation plants depicted in Figure 1 envelops all the engineering phases starting from the tendering process and ending with the maintenance of the system in normal operation mode.

The engineering of a new automation plant starts with the project initiation when a customer contacts a plant manufacturer and defines a specification sheet. At this point, the desired outcome is clear but the process is vaguely described.

In our envisioned engineering process the plant manufacturer designs an automation system serving the customer's requirements by using globally valid, reusable, and standardized automation components out of a library (R4). These automation components specify their general range of abilities by offering standardized skills [20] but do not contain any manufacturer specific nor plant topology relevant information (R1). Like presented in the SkillPro project by Julius et al. [21], AML has been used to represent manufacturing resource capabilities. However, a combination of several skills to a more complex skill is supported. Based on the product-process-resources-aspect (PPR) the plant sequence is specified by using these skills in a functionally oriented manner [6]. The design process is performed using an engineering tool which allows checking for valid combinations of different modules (R5.2). As a result, the modeled automation system can, at this point, already be used for rough simulation and proof of concept (R5.3). The plant model is saved and can be reused at a later point (R2).

After creating the concept, the customer contracts the plant

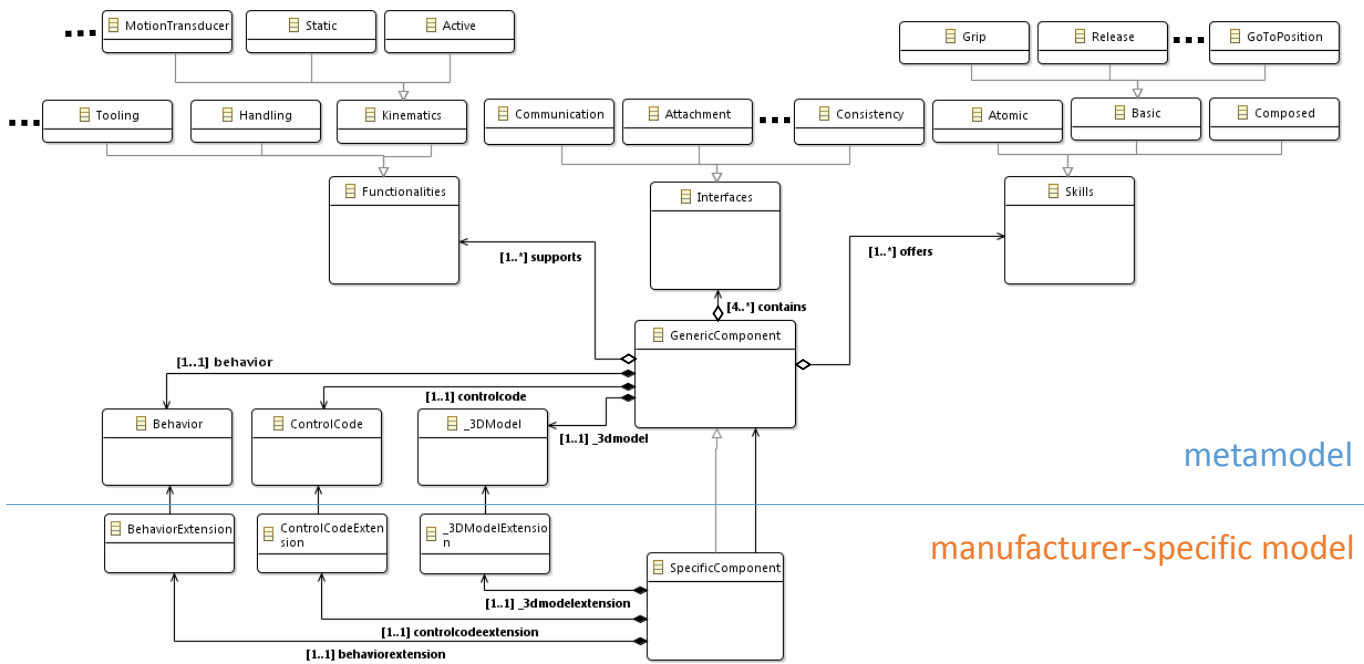


Fig. 2. Core part of the metamodel and relation to manufacturer specific model

manufacturer and makes the final selection of the automation components. The envisioned engineering approach aims at reusing the previously stored plant model (R2) and exchanging the general components with more specific components from different manufacturers. The replacement process is controlled by an engineering tool checking whether the respective components support the same functionality (R5.1). Afterwards the ordering of the components takes place.

Before delivery and assembly, the plant process flow is extended by the plant manufacturer by using the standardized skills and by programming special purpose tasks manually. Furthermore, the other crafts, such as electricians and mechanics, start their engineering activities based on the plant model (R2, R3) and save changes back into the plant model supporting cross-domain information so that a common data base is used (R3.1, R3.2). Parallel engineering is supported by the envisioned engineering process but is out of the scope of this paper. Initial parametrization, configuration, and tests based on the plant model are performed at this time to ensure that fewer errors occur when the automation components are delivered and assembled for the first time.

During commissioning, changes and optimization to the automation systems are documented in the cross-domain plant model. In normal operation mode, the plant model is equivalent to the constructed automation system and is used for monitoring and supervision purposes. Maintenance and extensions by different teams are first planned and simulated on the general plant model (R5.3) before being realized on the automation system.

C. Proposed metamodel

In order to comply with the aforementioned requirements and envisioned engineering approach, a model-based approach

was developed. The concept of reusing modules has been realized in the form of a metamodel depicted in Figure 2. The metamodel is used to derive modeling concepts and to create models of automation components at different abstraction levels. The proposed metamodel does not contain any plant topology information or reference and relation information between components. Furthermore, the metamodel is assumed to be universally valid and applicable to any plant engineering process in the automation domain. On the one side, the metamodel offers different classifying information like functionalities, interfaces, and skills depicted in the top part of Figure 2. On the other side, the metamodel comprises generic components which are modeled with the aid of implementation information like behavior, control code, and 3D-information.

The metamodel consists therefore of the following classes:

- **Functionalities** as a semantic description of a purpose of an automation component. A functionality (e.g. Active rotative element, Gripping element) can be assigned to a modeled component.
- **External/internal interfaces** for creating various links between and in modeled components. Automation components are linked together for modeling logical and physical communication paths, physical associations (e.g. gripper is attached at linear axis), and rule-based consistency checks (e.g. a gripper cannot be attached to another gripper). Further domain-specific links like pneumatic and electrical connections are also modeled. Moreover, component internal links between different domains can be created for modeling cross-domain dependencies such as a relation between a skill and the corresponding control code.

- Skills describing reusable production steps of automation components (e.g. *grip*, *release*). A modeled automation component can offer one or more standardized skills in order to allow functional engineering.
- Behavior models describing the desired behavior of automation components.
- Skeletons of control codes defining the structure of the control code. Specific control code is not available on the level of the metamodel.
- 3D-representations with kinematics of abstract automation components used for simulation and virtual commissioning.

The behavior, control code, and 3D-model are always directly related to a generic component and cannot exist on their own.

Based on the classes contained in the metamodel, generic components are created. Those generic components are universally valid, manufacturer independent, highly abstracted, and cross-domain models of automation components. The models of generic components are used for modeling an automation plant in early engineering phases and are classified following the approach presented by Helbig et al. [22]. This approach ensures a unique identification by applying an hierarchical classification and also supports the creation of shortcuts. Generic components are manufacturer independent and abstract. Therefore no explicit control code is available. Instead, only a rough structure with standardized code snippets is needed, e.g. program organization units (POU) in IEC 61131. The code skeleton guides the component manufacturer when implementing the actual control code. The 3D-models related to the generic components show simplified illustrations of the mechanical structure.

From the metamodel, models of manufacturer specific components inheriting from generic components are created as depicted in the bottom part of Figure 2. The manufacturer specific components can be enriched with additional information from the aforementioned classifying information. Furthermore, the implementation information, such as behavior, control code, and 3D-model, are extended at this point of modeling. By inheriting universally valid information from generic components, the replacement of generic components with specific components as mentioned in the envisioned engineering is realizable.

IV. APPLYING THE METAMODEL

Combining the envisioned engineering approach with the metamodel leads us to a tripartite model-based approach. Each stage builds on the predecessor and is served by one stakeholder.

The identified stakeholders in the engineering process of an industrial automation system and their tasks are depicted in Table I. Moving from the first to the last stakeholder corresponds to a more and more specific definition of the involved components in the final plant as each stakeholder expands the information defined by the predecessor. Our approach assumes that expert knowledge (e.g. programming and controlling a

specific component) is located and encapsulated on the level of the component manufacturer, but not any more on the level of the plant manufacturer. The latter uses standardized interfaces to interact with the component without in-depth knowledge of the exact functional principle.

TABLE I
STAKEHOLDERS AND THEIR ROLES

Stakeholder	Task	in-depth knowledge	abstraction level
Standardization committee	Definition of a metamodel containing semantic properties and interfaces used for description of generic components	none	high
Component manufacturer	Description of manufacturer-specific components based on metamodel	high	middle
Plant manufacturer	Creation of plant topology using generic/specific components	low	low

In the following, each stakeholder and their individual tasks are presented in more detail.

A. Standardization committee

The major task of the standardization committee is offering a metamodel that contains a set of generic components describing automation components on a high abstraction level. In-depth knowledge is not available in this modeling stage. Generic components are created out of composite elements from the other libraries stored in the metamodel. These libraries include information concerning functionalities, skills, interfaces, behavior, control code, and 3D-information. A basic generic component such as a *GenericGripper* will therefore consist of:

- a functionality *Gripping* element
- several external interfaces for a mechanical attachment, a data- and a power-link, and internal interfaces for linking different subdomains together (e.g. POU *closeGripper* in control code is linked to skill *grip* which is linked to kinematics *moveJaws(together)* in 3D-model)
- the skills *grip* and *release*
- a behavior model with two states *closed* and *opened* and transitions
- a skeleton of control code with two predefined POU's *closeGripper* and *openGripper*
- a 3D-model with kinematics showing an abstract component with two claws

The top left part of Figure 3 depicts the composition of the *GenericGripper*.

B. Component manufacturer

Based on the metamodel, the component manufacturer inherits the predetermined structures from generic components

and extends them with additional information. An adaptation of the predefined structures is not possible as it would contradict the idea of inheritance. Figure 3 shows in the center row a manufacturer specific component *SpecificGripper*. It inherits all the information from the *GenericGripper* and extends predefined sub-areas of the *GenericGripper* with concrete 3D-information, control code, and behavior description. At this point, the manufacturer-specific knowledge comes into play as this stakeholder knows best how to control the component and how it should behave. Therefore, the explicit control code for opening and closing the jaws is written and embedded in the corresponding POU's of the code skeleton of the generic component. Furthermore, the actual 3D-model of the *SpecificGripper* extends the abstract 3D-model of the *GenericGripper*. If necessary, additional internal links between domain specific information are also created using internal interfaces to enable a cross-domain modeling of dependencies in the automation component. Finally, the component manufacturer saves his newly created manufacturer-specific models of components in a separate own library which is placed at the disposal of the last stakeholder.

C. Plant manufacturer

The last engineering step of designing the automation system as presented in Section III-B is performed by the last stakeholder, the plant manufacturer. In the presented approach, the plant manufacturer has the option of choosing the desired components out of a pool of manufacturer-specific libraries from the second stakeholder containing the range of offered products. All presets of the component manufacturer, and therefore also from the standardization committee, are taken over to offer the modeled component as a ready-to-use component to the plant manufacturer. The plant manufacturer no longer needs any in-depth knowledge in this engineering phase as controlling the component is done on a functional level. Adaptations to the control code or behavior can only be done if the component manufacturer allows such changes. Additionally, the plant manufacturer also has the option to model the automation system with generic components out of the metamodel in a conceptual way.

After having specified the different components in use, the plant manufacturer models his (special purpose) machine and defines the PPR-aspects in the plant model. Furthermore, this stakeholder is responsible for the creation of the overall sequence flow of the plant by using the functionalities offered by the components and combining them to process steps.

The plant model is finally given to the plant engineer who is the end-user. Based on the plant model he has the possibility to perform simulations and coordinate maintenance tasks.

V. SUPPORTING THE ENGINEERING APPROACH IN AUTOMATIONML

For implementing the presented concept, different data exchange formats for modeling automation components, sequences, and for supporting an ontology have been analyzed. The choice fell on the markup language AutomationML,

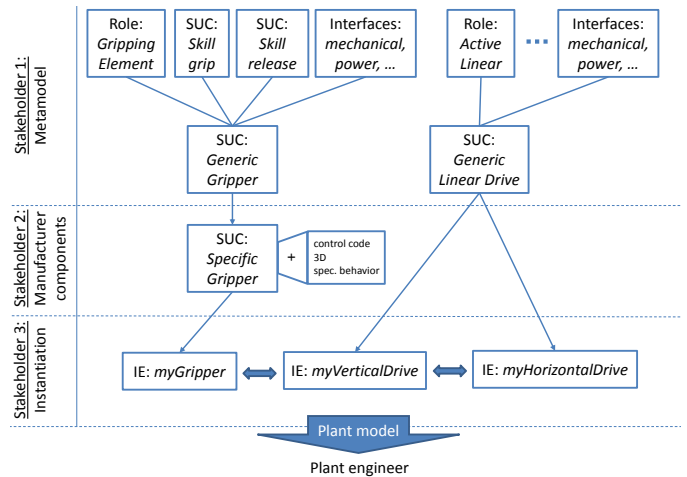


Fig. 3. Exemplary relation of the presented elements implemented in AML an emerging data exchange format and standard used for continuous data management throughout the different stages of engineering. AML enables the modeling of domain-specific information.

A. Metamodel in AutomationML

The envisioned engineering approach is based on the metamodel presented in Section III-C. Each class of the metamodel is implemented in an AML-specific format and is depicted in the top row of Figure 3.

Functionalities of automation components are AML-roles collected in a RoleClassLibrary and are categorized as presented in [22]. Different functionalities describing a type of components such as an active linear element, a spring, a vane or a gripping element in a semantic way can be simultaneously assigned to a modeled component in order to represent its diversity.

The internal and external interfaces of the metamodel are realized in AML as interfaces. On the one side, more specific interfaces inherited from the *AttachmentInterfaces* are modeled in order to allow a rule-based assembly of components. A component with a flange type A can therefore not be fixed to another component with a flange type B. On the other side, the *Communication* interface stored in the AutomationMLBaseInterface library is extended with two new classes: PhysicalCommunicationEndpoints containing different types of plugs (e.g. Ethernet, RS232, USB-A, ...) and LogicalCommunicationEndpoints allowing logical connections between interfaces (e.g. Master, Slave, Bus, ...). The *AttachmentRules* interfaces contain all the classes listed in [22]. Those interfaces can be used to model valid physical combinations of components. Subsequently, a consistency check, as presented in [23], can be performed to support the engineer during the engineering of the automation plant. The consistency check verifies the combination of automation components and displays unsupported connections of components such as fixing a gripper to another gripper. A supported combination of components is modeled by assigning the same AttachmentRule interface to different components and therefore allowing a link between

both components.

A `SystemUnitClass` (SUC) containing skills is also offered by the metamodel. Different skill concepts have already been presented in [20], [24] and a combination of these concepts is modeled in the new SUC *Skills*. This SUC is not inheriting from predefined AML elements but is used to enable functional engineering [25]. It means that skills are assigned to generic and specific components while modeling the components and these skills are called afterwards by the engineering platform to perform a component-specific action. A skill modeled in the AML metamodel consists of the semantic description of the component-specific action and a *LogicInterface* pointing to a specific POU situated in a separate PLCopen XML file containing the skeleton of the control code.

AML offers a simple way for linking the behavior and control code to a generic component by using *PLCopenXML-Interfaces*. These interfaces point to external PLCopen XML files containing structured code in IEC 61131 that are used for modeling behavioral information and for creating control code skeletons. A *COLLADAInterface* is used analogously for referring to an external Collada file containing 3D- and kinematic information.

By combining the elements of the AML metamodel described above, the creation of generic components in SUCs is possible. They are categorized in a tree structure as presented in [22] and inherit all modeled information from their father elements. The main SUC containing all the generic components covers all different types of components in the domain of industrial automation by specifying different superclasses like kinematics, handling, and tools, down to a certain level in a tree structure. As soon as manufacturer specific knowledge is needed, the generic component cannot be specialized any more. Generic components modeled as SUC in AML are e.g. a *GenericLinearDrive* which can be found in the categories *KinematicComponents-ActiveKinematicComponents-ActiveLinearComponents* or a *GenericGripper* located in *HandlingComponents-GrippingComponents*.

B. Modeling of manufacturer-specific components in AutomationML

Offering the AML metamodel to the second stakeholder enables the creation of his own manufacturer-specific models of components and organizing them in his own manner. This stakeholder creates a SUC library containing specific components modeled as SUCs which inherit from generic components as depicted in the middle row of Figure 3. The referenced PLCopen XML files of the generic components containing the behavior and control code are extended with more specific structured code. If necessary, the newly created SUCs are expanded with additional information offered by the metamodel (e.g. additional Skills-SUCs) or by the AutomationMLBase classes (e.g. PLCopenXMLInterface). The expansion is achieved by instantiating the corresponding SUC needed inside the SUC of the specific component.

Finally, the manufacturer-specific components are stored in a separate library as SUCs and are exported afterwards as

InternalElements (IE). By offering the IEs to the third and last stakeholder, this stakeholder is able to model an individualized automation system with manufacturer specific components or with generic components from the metamodel as depicted in the bottom row of Figure 3.

VI. EVALUATION

To evaluate the proposed approach, a simplified application scenario considering only the automation components of an educational purpose pick and place unit is presented here. As described earlier, it is assumed that the plant manufacturer and the plant engineers don't need to have minimal in-depth knowledge of the automation components. The procedure explained here encompasses the whole engineering process involved for the pick and place unit.

The process starts when the plant engineers tender their functional requirements, which are depicted as stage 1 in Figure 4. Here the functional requirements are *Grip/Release Workpiece*, *Linear Movement-Vertical*, and *Linear Movement-Horizontal* in order to form a pick and place unit. The plant manufacturing companies complying to the presented standard are able to create a pick and place unit at this stage with generic components as envisioned in Section III-B. Here the functional requirement *Linear Movement-Vertical* and *Linear Movement-Horizontal* is realized by generic components which offer the functionality `Active linear element`. The requirement *Grip/Release Workpiece* is realized by a generic component which offers the functionality `Gripping element`. The pick and place unit developed at this stage is already capable of storing behavioral information, basic CAD models, and communication interfaces. A rough simulation of the pick and place unit is also possible at this stage. The engineering groups involved and their interdependencies can also be defined. The data model implemented in AML permits this plant model and the complete engineering data to be exchanged among the engineering groups and thus facilitates parallel engineering.

Satisfied with the generic pick and place unit, the plant engineers work on detailed specifications of the functional components to accommodate the unit in their plant in stage 2. This includes the following specifications: 1) *overall repetition accuracy* of 0.5 mm, 2) *vertical stroke* of 70 mm 3) *horizontal stroke* of 120 mm and 4) *pay load* of 50 g. The plant manufacturer maps these specifications on to the individual components. Here the *vertical stroke* is applied to the *GenericLinearDrive* with the functionality `Active linear element` used in the vertical axis, *horizontal stroke* is mapped onto the *GenericLinearDrive* with the functionality `Active linear element` in the horizontal axis and the *pay load* is mapped onto the *GenericGripper* with functionality `Gripping element`. In order to achieve the *overall accuracy*, the interrelations among the detailed parameters are analyzed and corresponding changes are initiated.

In stage 3, the plant manufacturer takes a list of component manufactures, who supply their components complying to the metamodel standards, in order to make the final selection and

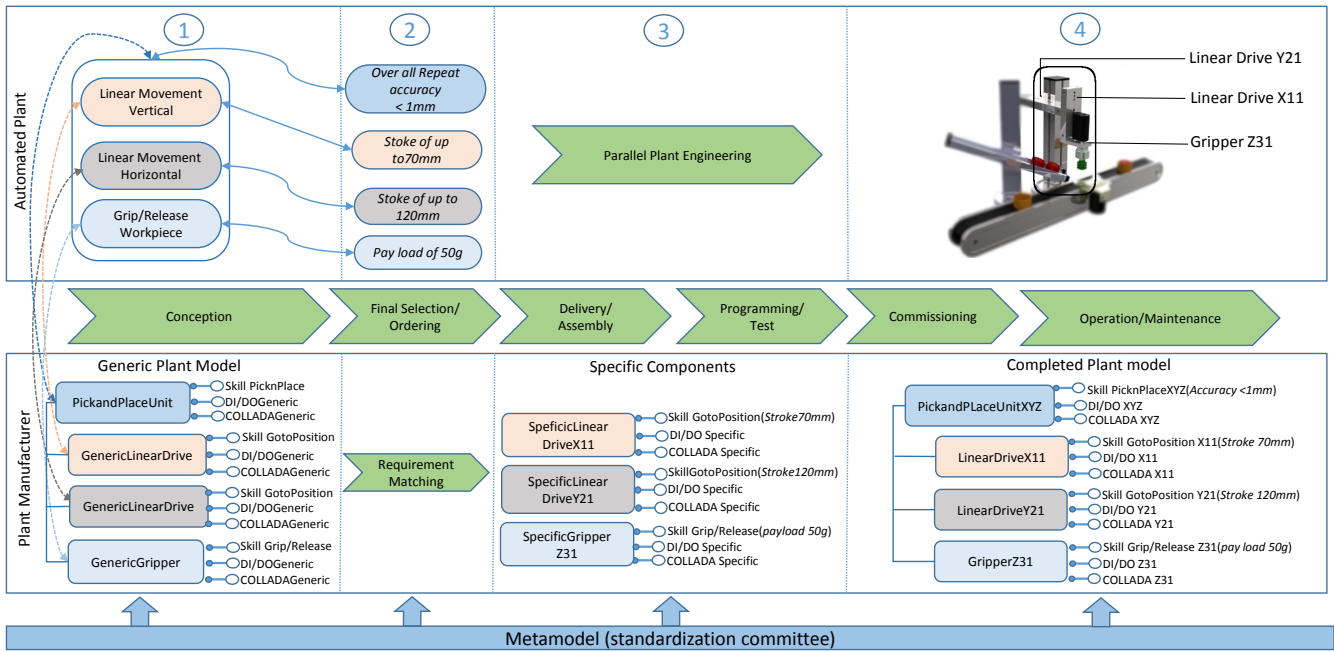


Fig. 4. Engineering of a pick and place unit

ordering. The AML engineering data at this stage allows the detailed specifications to be distributed among the different engineering groups and execute detailed engineering process in parallel. The specific components built are embedded with a detailed engineering data such as completed control code in PLCopen XML, CAD in Collada format etc. The specific components provide standardized interfaces complying to the presented approach.

With readily available specific components containing detailed engineering data, the plant manufacturer commissions the plant with minimum effort as depicted in stage 4. The completed plant model in AML can be used as standard for data exchange and coordination for further operation and management of the plant.

By following the above engineering process, the following improvements to the conventional engineering process have been observed.

- Involvement of a standardization committee reduces the gap between the different stakeholders involved in the process.
- The use of generic components enables maximum reuse of components with minimum effort in engineering.
- The semantic gap between different stakeholders is minimized.
- The use of AML as a data exchange format helps for interdisciplinary interaction and integration.

VII. CONCLUSION

This paper addresses one of the key requirements of a manufacturing company in the modern scenario: its ability to design, test and market high quality and low cost products in a short time frame. A model-based engineering approach is presented which reduces overall development

efforts with an engineering data model and a standardized engineering process. The engineering process is standardized with a tripartite process based on an engineering data model. The data model facilitates the coordination of the different engineering groups involved in the process, tool integration and data exchange. The concept introduces an engineering approach which relocates the required expertise and reduces the chances of modeling errors by introducing a metamodel which offers modularity, reuse of components, and variant management. The proposed idea is evaluated by implementing the concept in the automation data exchange standard AML. The results show significant improvement compared to the conventional engineering process by reducing the gap between different stakeholders involved in the engineering process of a manufacturing plant and promoting the interaction among different engineering groups.

Additional studies are being performed to further analyze the applicability of the approach. A component generator tool which generates an AML description of specific components out of generic components is completed. The engineering tool support for the above proposed model is analyzed with an importer tool which converts the AML engineering data model to an IEC 61131-3 standard control program. Furthermore, the import of the model to other engineering tools is also planned.

ACKNOWLEDGMENT

The concept presented in this paper is developed as a part of the research project OPAK (Funding no. 01MA13012F). This work has been supported by the German Federal Ministry of Economic Affairs and Energy (BMWi) as part of the research program "Autonomik Industrie 4.0". The authors are responsible for the content of this publication.

REFERENCES

- [1] G. Da Silveira, D. Borenstein, and F. S. Fogliatto, "Mass customization: Literature review and research directions," *International journal of production economics*, vol. 72, no. 1, pp. 1–13, 2001.
- [2] A. Bardakci and J. Whitelock, "How ready are customers for mass customisation? an exploratory investigation," *European Journal of Marketing*, vol. 38, no. 11/12, pp. 1396–1416, 2004.
- [3] J. Jiao, Q. Ma, and M. M. Tseng, "Towards high value-added products and services: mass customization and beyond," *Technovation*, vol. 23, no. 10, pp. 809–821, 2003.
- [4] Q. Tu, M. A. Vonderembse, and T. Ragu-Nathan, "Manufacturing practices: antecedents to mass customization," *Production Planning & Control*, vol. 15, no. 4, pp. 373–380, 2004.
- [5] R. Isermann, "Mechatronic systemsinnovative products with embedded control," *Control Engineering Practice*, vol. 16, no. 1, pp. 14–29, 2008.
- [6] F. Himmler, "Function based engineering with automationml - towards better standardization and seamless process integration in plant engineering," *Wirtschaftsinformatik Proceedings*, no. 2, 2015.
- [7] B. Vogel-Heuser, C. Legat, J. Folmer, and S. Rösch, "Challenges of parallel evolution in production automation focusing on requirements specification and fault handling," *at-Automatisierungstechnik*, vol. 62, no. 11, pp. 758–770, 2014.
- [8] G. Hackenberg, C. Richter, M. F. Zaeh, D. Ascher, P. Stich, G. Reinhart, S. Teufl, A. Campetelli, C. Legat, J. Mund *et al.*, "From conception to refinement in mechatronics systems engineering," in *Proceedings of the International Conference on Mechatronics and Robotics Engineering, Kuala Lumpur*, 2015.
- [9] R. Drath, A. Lüder, J. Peschke, and L. Hundt, "Automationml-the glue for seamless automation engineering," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2008, pp. 616–623.
- [10] S. Feldmann, C. Legat, and B. Vogel-Heuser, "An analysis of challenges and state of the art for modular engineering in the machine and plant manufacturing domain," *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 87–92, 2015.
- [11] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, "A model-based method for organizing tasks in product development," *Research in Engineering Design*, vol. 6, no. 1, pp. 1–13, 1994.
- [12] P. Ovtcharova, I. Jivka, M. Marinov, D. Gutu, D. Szots, A. Simonyi *et al.*, "Representation of cross-domain design knowledge through ontology based functional models," in *DS 68-6: Proceedings of the 18th International Conference on Engineering Design (ICED), Impacting Society through Engineering Design, Lyngby/Copenhagen, Denmark*, vol. 6: Design Information and Knowledge, 2011.
- [13] VDI/VDE, "Guideline 3695, engineering of automation systems, section 1," 2010.
- [14] L. Berardinelli, S. Biffi, E. Maetzler, T. Mayerhofer, and M. Wimmer, "Model-based co-evolution of production systems and their libraries with automationml," in *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. IEEE, 2015, pp. 1–8.
- [15] N. Jazdi, C. Maga, and P. Göhner, "Reusable models in industrial automation: experiences in defining appropriate levels of granularity," in *Proc. of the 18th IFAC World Congress*, 2011, pp. 9145–9150.
- [16] O. Graeser, "Durchgängige datenhaltung schafft wettbewerbsvorteil," *Maschinenmarkt*, vol. 30, p. 31, 2014.
- [17] K. Thramboulidis, "Overcoming mechatronic design challenges: the 3+1 sysml-view model," *Computing Science and Technology International Journal*, vol. 1, no. 1, 2013.
- [18] S. J. Herzig, A. Qamar, A. Reichwein, and C. J. Paredis, "A conceptual framework for consistency management in model-based systems engineering," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2011, pp. 1329–1339.
- [19] B. Vogel-Heuser, C. Diedrich, A. Fay, S. Jeschke, S. Kowalewski, M. Wollschlaeger *et al.*, "Challenges for software engineering in automation," *Journal of Software Engineering and Applications*, 2014.
- [20] M. Bengel, "Modelling objects for skill-based reconfigurable machines," *Pham, D. T. et al. (Edit.): Innovative production machines and systems*, pp. 238–243, 2008.
- [21] J. Pfrommer, M. Schleipen, and J. Beyerer, "Pprs: Production skills and their relation to product, process, and resource," *IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, 2013.
- [22] T. Helbig, S. Henning, and J. Hoos, "Efficient engineering in special purpose machinery through automated control code synthesis based on a functional categorisation," *machine learning for cyber physical systems*, 2015.
- [23] T. Helbig, J. Hoos, and E. Westkämper, "Consistency check of the functional solution model in special purpose machinery," *49th CIRP Conference on Manufacturing Systems (CIRP-CMS)*, 2016.
- [24] J. Backhaus, M. Ulrich, and G. Reinhart, "Classification, modelling and mapping of skills in automated production systems," *5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV)*, 2013.
- [25] C. Sünder, A. Zoitl, and C. Dutzler, "Functional structure-based modelling of automation systems," *International Journal of Manufacturing Research (IJMR)*, 2006.