Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 17 (2014) 451 – 456
www.elsevier.com/locate/procedia

Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems

Combining a SysML-based modeling approach and semantic technologies for analyzing change influences in manufacturing plant models

Stefan Feldmann*, Konstantin Kernschmidt, Birgit Vogel-Heuser

Institute of Automation and Information Systems, Technische Universität München, Boltzmannstr. 15, 85748 Garching near Munich, Germany

* Corresponding author. Tel.: +49 89 289 16441; fax: +49 89 289 16410. *E-mail address:* feldmann@ais.mw.tum.de

Abstract

During the lifecycle of mechatronic manufacturing systems, repeatedly exchanges of different system elements have to be conducted. These system elements can be either single discipline-specific components, or modules composed of components or further modules. In order to ensure the elements' compatibility and the system's functionality, which must comply with the specification after the exchange, a model-based analysis of the change influences is presented in this paper. A SysML-based modeling approach is combined with the formal representation of the model in an OWL ontology to conduct the required compatibility check. By that, the disciplines involved in the engineering process, e.g. mechanics, electrics/electronics and software, can be modeled and taken into account for the analysis of change influences. Thus, this paper contributes to the domain of manufacturing systems by providing a meta model for the interdisciplinary modeling of manufacturing systems on the one hand and by defining a corresponding formal representation intended to ensure system elements' compatibility on the other hand.

Selection and peer-review under responsibility of the International Scientific Committee of "The 47th CIRP Conference on Manufacturing Systems" in the person of the Conference Chair Professor Hoda ElMaraghy"

Keywords: Manufacturing systems; Model-driven development; Semantic technologies; Validation; Compatibility check

1. Introduction

Engineering of manufacturing systems is a complex process involving experts from multiple disciplines, e.g. mechanical, electrical and software engineering. During its lifecycle, a manufacturing system evolves through changes of system elements, e.g. due to requests for increased product variety [1]. These system elements can be either single discipline-specific *components*, e.g. bus clamps, or *modules* composed of components or further modules, e.g. a cylinder. To cope with the system's complexity, it is indispensable to ensure its correct functionality [2]. One challenge is to ensure the elements' compatibility after a change, as interfaces between connected parts of the system must match and required functionality must be provided. Another challenge is to ensure, that the manufacturing system's variables fulfil the imposed performance requirements [3]. To make things worse, in a mechatronic manufacturing system, an exchange of system elements can have discipline-specific as well as interdisciplinary influences on other elements, e.g. for a sensor exchanged, electrical connections and the software can be affected. To consider these influences, an interdisciplinary system model and an analysis of the elements' compatibility is needed. The Systems Modeling Language (SysML) enables the comprehensible modeling of manufacturing systems [4]. SysML however lacks a formal base, and, thus, compatibility checks cannot be performed directly [5]. An approach that enables the explicit knowledge representation and the application of inference mechanisms is the Web Ontology Language (OWL). However, OWL is intended for a formal knowledge representation and, thus, lacks in comprehensible modeling. Therefore, this paper proposes to combine systems modeling with semantic technologies, i.e. OWL, for performing compatibility analyses and, hence, change influence analyses. The approach is exemplarily applied for SysML4*Mechatronics* [6], which focuses on the discipline-specific and interdisciplinary connections of system elements and their port specifications.

Selection and peer-review under responsibility of the International Scientific Committee of "The 47th CIRP Conference on Manufacturing Systems" in the person of the Conference Chair Professor Hoda ElMaraghy"
doi:10.1016/j.procir.2014.01.140

The paper is organized as follows: In the next section, related work on systems modeling and semantic technologies is discussed. Subsequently, an application example motivating the proposed approach is introduced. The proposed approach combining SysML4*Mechatronics* and OWL for applying compatibility analyses as well as its evaluation using a lab-size demonstration model are presented in sections 4 and 5, respectively. Finally, the paper is concluded in section 6.

2. Related work

The Systems Modeling Language (SysML) is a graphical modeling language for systems engineering [4]. In order to integrate disciplines of the engineering process with their own approaches within the modeling notation, Thramboulidis [7] proposes the application of SysML profiles for realizing a 3+1 view model. A similar approach for integrating discipline-specific views was proposed in [8] through applying profiles for modeling and model transformations for consistency checks between the views. The application of SysML as high-level modeling notation was proposed in [4]. Besides a methodology for engineering the system, the authors propose mechanisms for consistency check and validation of models. Mechatronic compatibility checks however were not proposed by these approaches yet. Nevertheless, due to the semi-formal character of SysML, formal models and computer-interpretable mechanisms are needed to implement compatibility checks [5]. AutomationML addresses the data exchange between disciplines and along the life cycle of a production system for ensuring interoperability between tools within manufacturing systems engineering [9]. However, in contrast to SysML, AutomationML does not inherently provide standardized graphical modeling support within appropriate diagrams to model the systems structure, behavior and requirements. Graphical tools and appropriate interfaces need to be implemented for AutomationML for making it applicable to the interdisciplinary manufacturing systems domain. In [10] such an architecture for collaboration between tools and persons within the engineering process is proposed. However, the means for ensuring consistency within the models, e.g. using a formal knowledge base, is not provided, which is addressed in this paper.

While classical programming approaches require to code knowledge explicitly within the program code, knowledge-based systems enable the explicit representation of knowledge as well as its automatic processing [11]. An established technology for formal knowledge representation is the Web Ontology Language (OWL). Using standardized, deductive reasoning mechanisms, in OWL formulated ontologies can be processed automatically e.g. to identify inconsistencies within the model. Further mechanisms, e.g. query languages (SPARQL Protocol and RDF Query Language, SPARQL), enable the extension of OWL's expressiveness. Recently, ontologies are applied for semantic description of requirements imposed on building automation systems [12] as well as for device description for verifying the consistency, completeness and correctness of devices and the system's design [13]. Lastra et al. [14] extend the IEC Automation Object Reference through a mechatronics ontology and an IEC 61499 ontology for ensuring reusability and interoperability of automation devices. In Lohse et al. [15], a device ontology according to the function-behavior-structure paradigm was developed for selecting, configuring and evaluating system alternatives. However, all these approaches do not focus a visual modeling notation for modeling the production system. Hence, the use of CAEX for modelling the plant model, its validation using OWL [16] and the identification of inconsistencies and redundancies within the CAEX model [17] was introduced. Nevertheless, these works neglect the holistic mechatronic structure of an automation system. Another approach addressing the management of model versions by consistency-preserving merges for several modeling notations using OWL was presented by [18]. The applicability for production system's compatibility analysis needs to be evaluated. Further approaches focus on verifying behavioral aspects of manufacturing systems, e.g. mode handling of flexible manufacturing systems [19], but do not consider the structural compatibility of the manufacturing system's elements.

Concluding, various approaches for modeling the interdisciplinary aspects of a production systems exist. Due to their semi-formal nature, processing of the modeled knowledge, e.g. checking the compatibility of model elements, is not supported inherently. Thus, the application of formal knowledge representation and processing of the modeled knowledge is needed. Within this area, numerous approaches for enabling compatibility check and retrieval of modeling elements exist but neglect their comprehensible modeling. An approach that integrates a modeling notation for interdisciplinary systems and formal knowledge representation for verifying compatibility of modeling elements has been not proposed, yet.

3. Application example: Pick and Place Unit

For the described approach, a bench-scale model of a production process serves as an application example. For the sake of clarity and simplicity, the application example was reduced to the substantial facts for illustrating the approach. The Pick and Place Unit (PPU) [20, 21] consists of a stack depot, a crane, a stamp module and a sorting belt. The stack depot poses as material source of the process. In the first step, a work piece (WP) is pushed into the handover position. The crane grabs the WP at 0° and transports the WP to the stamp located at 180°. After the crane has released the WP it is clamped, stamped and released afterwards. In the last step, the crane transports the WP to the sorting belt at 90°. Finally, the WP is transported, according to its color and material, to the appropriate slide (Fig. 1).

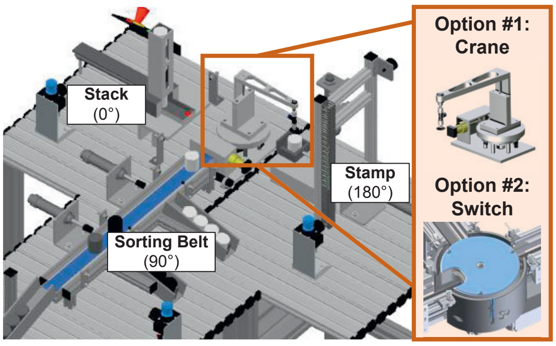

Fig. 1. CAD-model of the Pick and Place Unit [20, 21].

Thus, the crane constitutes a central module to fulfill the functionality “Move WP circularly”. In order to fulfill this functionality, the crane module consists of the mechanical crane structure, a cylinder to lift the crane, a turntable to fulfill the rotational movement, a vacuum gripper to grab the work pieces and an according software component to control the crane. However, a crane is not the only possible solution to fulfill the functionality “Move WP circularly”; instead also a switch with conveyor belts at the according positions (0°, 90°, 180°) could be utilized, which uses a different operating principle to achieve the same outcome. A changed customer requirement, e.g. that the work piece should not be lifted up, could make such an exchange of the module necessary.

If such an exchange shall be implemented to the production system, it is essential that all influences have been taken into account. In order to analyze, which influences an exchange of the crane through a switch would have on the existing system, it has to be verified which interfaces are compatible and which interfaces have to be changed in the different disciplines. Therefore, an integrated system model (SysML4*Mechatronics*), including the components and ports of each discipline, and an according formal model (ontology) will be utilized.

4. Approach for integrating model-based engineering and semantic technologies

In the following, the approach for integrating model-based engineering and semantic technologies for applying compatibility checks (cf. Fig. 2) is presented for the PPU. Using SysML4*Mechatronics*, the system can be modeled (section 4.1). Subsequently, the modeled information is transformed into the OWL ontology (section 4.2). Using SPARQL queries, incompatibilities can be identified (section 4.3).

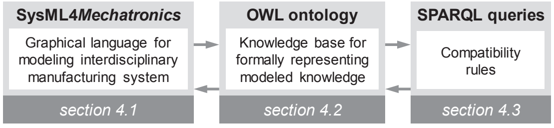

Fig. 2. Overview on the proposed approach

4.1. SysML4Mechatronics model

In the first step during the development phase, the mechatronic production system is modeled in SysML4*Mechatronics*. Fig. 3 and Fig. 4 show parts of the meta model. Elements of each discipline (Mechanical-Block, E/E-Block, and Software-Block) are included into the model during the design and connected with according ports (Mechanical-Port, E/E-Port, Software-Port). Thus, through the integrated model, elements and interfaces can be described discipline-specifically as well as interdisciplinary.

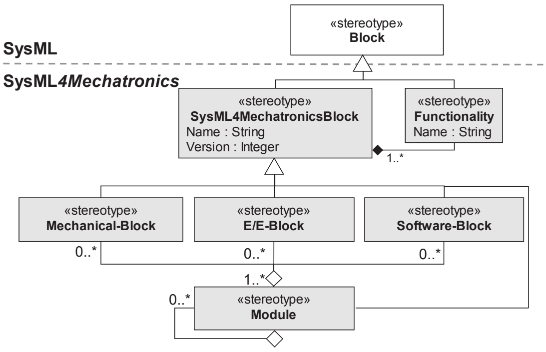

Fig. 4. Part of the SysML4*Mechatronics* meta model (block, module definition).

Fig. 5 shows e.g. the crane and the switch modules as black boxes with the ports resulting from their internal components in the different disciplines. Due to spatial restrictions, the modules’ port connections to corresponding ports of other modules are not shown in the figure. While the E/E-components of the crane module are connected directly to the digital binary in- and outputs (24 V DC) of the Programmable Logic Controller (PLC), the switch module is connected through a fieldbus, i.e. Profibus DP, to the PLC. Furthermore, blocks and modules have functionalities, which they fulfill (e.g. an inductive sensor fulfills the functionality “detect metal work pieces”). In order to implement a specific functionality, certain ports are mandatory (e.g. the power supply for a motor is mandatory for each functionality, while the port for pulse width modulation is only mandatory for the functionality “velocity control”).

After the system has been modeled in the described way during the development, change influences can be analyzed therein. In order to ensure a correct operation after the exchange of a system component or module, the required interfaces as well the functionality have to be fulfilled appropriately. Therefore, compatibility rules apply for the ports in each discipline, cf. section 4.3. In order to check the compatibility of the exchanged element with the existing system, a formal representation of the system is required, as described in the next section.

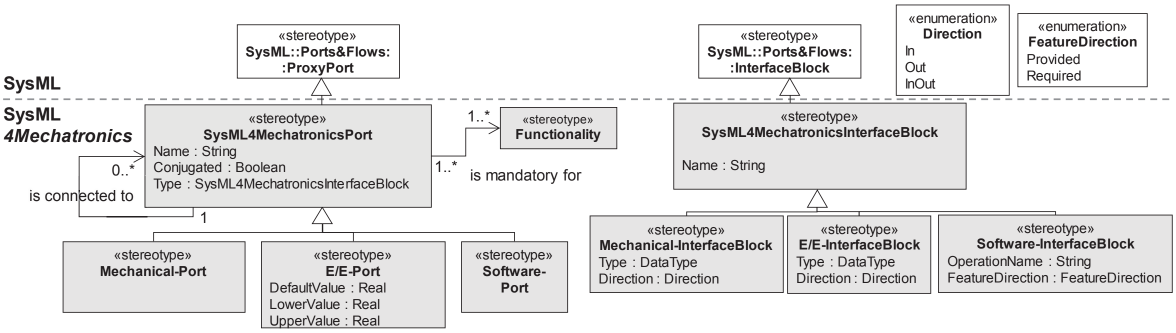

Fig. 3. Part of the SysML4*Mechatronics* meta model (Port definition).

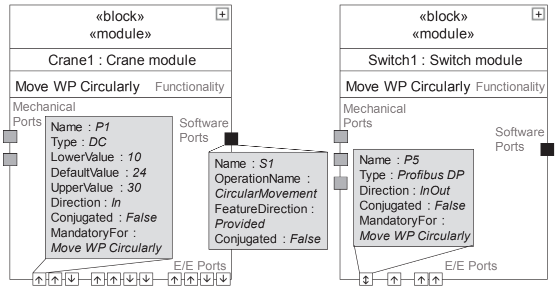

Fig. 5. Comparison of the crane module and switch module showing exemplarily the port specification of an E/E-port in each module.

4.2. OWL ontology

For the intended compatibility check, the SysML profile serves as the meta model, which will be represented within the terminological knowledge, thus, providing the base vocabulary of the ontology. The resulting model itself is then transformed into the assertional knowledge. The main concepts and relations of the OWL ontology are shown in Fig. 6. The stereotypes within the SysML4*Mechatronics* profile are transformed into corresponding OWL concepts, e.g. *SysML4MechatronicsBlock*, *-Port* and *-InterfaceBlock*. By that, specific modeling elements of a SysML4*Mechatronics* model are represented as OWL instances of these concepts. Associations modeled in between those modeling elements are represented as OWL properties. OWL literals are defined for respective attributes.

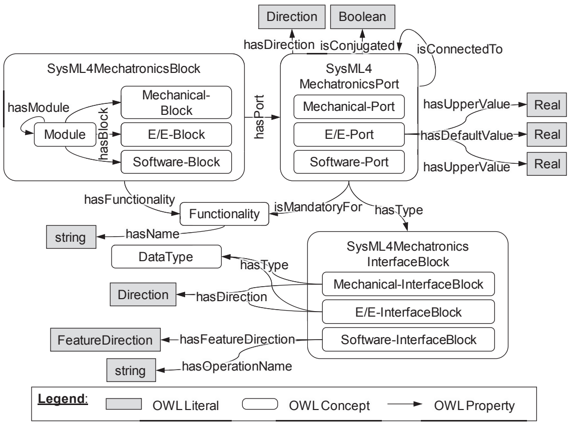

Fig. 6. Excerpt of the OWL ontology.

An exemplary representation of a model within the ontology's assertional knowledge for the module *Crane1* is shown in Fig. 7. The modeling elements of the SysML*4Mechatronics* model in Fig. 5 are represented using OWL instances of the OWL concepts, e.g. *Crane1* and *P1*, and related to each other via the OWL properties, e.g. *hasBlock* and *hasPort*. By that, the SysML4*Mechatronics* modeling elements have an equivalent representation within the ontology.

4.3. Compatibility rules for analyzing change influences

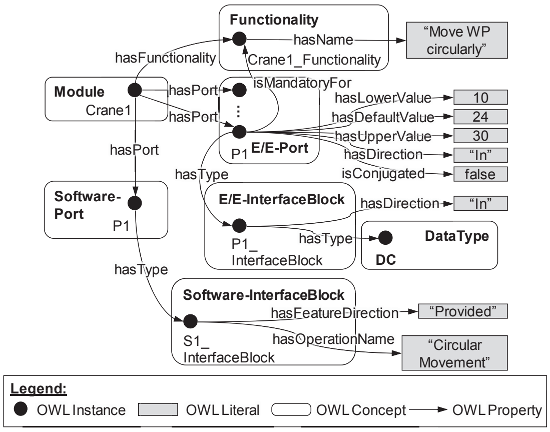

Fig. 7. Exemplary OWL representation of the crane example.

In order to analyze change influences in the system model, compatibility rules between exchanged elements need to be formulated. These compatibility rules are represented within SPARQL queries, which provide the opportunity to describe patterns that are retrieved from the OWL ontology. By that, specific incompatibility criteria can be formulated and results of the queries return instances that do not comply to the compatibility rule. Typical compatibility rules that apply for the SysML4*Mechatronics* model as well as their SPARQL formulation are shown in Table 1 and described in the following.

The first compatibility rule for mechanical and E/E-ports is the compliance of the connected ports' *data types*. Those types need to be exactly the same for two modules (variables *m1* and *m2*). Using the SPARQL query (1), all ports that are connected to each other (variables *p1* and *p2*) are queried. The query result is filtered if the connected ports' types (variables *t1* and *t2*) are not equal. In the application example, *P1* must be connected to a corresponding *DC* port and *P5* to a corresponding *Profibus DP* port (cf. Fig. 5). Another restriction, which has to be fulfilled by connected mechanical and E/E-ports, is their *direction* property. In-ports may only be connected to either Out-ports or InOut-ports, but not to In-ports (same applies for Out-ports). Thus, SPARQL query (2) is applied: If all ports are connected properly, the query returns no result, as e.g. the case for *P1* and *P5*; thus, their directions match the rule.

A further compatibility rule that needs to be fulfilled within system design is the fulfilment of connected ports' data *ranges*. In general, the range of an In-port must include the range of the Out-port it is connected to, resulting into several restrictions for the In- and Out-ports' lower, upper and default values. Within the SPARQL query (3), the first *FILTER* expression ensures that variable *p1* is the input of the connected port pair; the second one retrieves the results not compliant to the rule. By that, it is ensured that the Out-port's data range fits the In-port's acceptable range, i.e. that variable *p2*'s lower value *low2* is greater than or equal to variable *p1*'s lower value *low1* (definition of other values respectively). A detailed view of the connection between the crane's port *P1* and the PLC's port *P1_PLC* is shown in Fig. 8. In the illustrated example, the modules are connected correctly as the values of the Out-Port (*P1_PLC*) fit the In-Port's (*P1*) range. Similarly to mechanical and E/E-ports' data types, it must be ensured that software ports connected to each other refer to the same software *operation*. In the application example, the crane's software port *S1* refers

Table 1. Compatibility rules and corresponding SPARQL query.

Mechanical-Ports, E/E-Ports

Data Type – Same data types

```
SELECT ?m1 ?m2 ?p1 ?p2 WHERE {
    ?m1 :hasPort ?p1 . ?m2 :hasPort ?p2 . ?p1 :isConnectedTo ?p2 .
    ?p1 :hasType [ :hasType ?t1 ] . ?p2 :hasType [ :hasType ?t2 ] .
    FILTER ( ?t1 != ?t2 ) . }                                                    (1)
```

Direction – In – Out, InOut – InOut, In – InOut, Out – InOut

```
SELECT ?m1 ?m2 ?p1 ?p2 WHERE {
    ?m1 :hasPort ?p1 . ?m2 :hasPort ?p2 . ?p1 :isConnectedTo ?p2 .
    ?p1 :hasDirection ?d1 . ?p2 :hasDirection ?d2 .
    FILTER ( ?d1 = ?d2 && ?d1 != ”InOut” && ?d2 != ”InOut” ) . }                 (2)
```

E/E-Ports

Range – Range of In-Port must match range of Out-Port

```
SELECT ?m1 ?m2 ?p1 ?p2 WHERE {
    ?m1 :hasPort ?p1 . ?m2 :hasPort ?p2 . ?p1 :isConnectedTo ?p2 .
    ?p1 :hasDirection ?d1 . ?p2 :hasDirection ?d2 .
    ?p1 :hasLowerValue ?low1 . ?p2 :hasLowerValue ?low2 .
    ?p1 :hasUpperValue ?upp1 . ?p2 :hasUpperValue ?upp2 .
    ?p1 :hasDefaultValue ?def1 . ?p2 :hasDefaultValue ?def2 .
    FILTER ( ?d1 = ”In” || ?d1 = ”InOut” ) . FILTER ( ?low2 < ?low1 ||
    ?upp2 > ?upp1 || ?def2 < ?low1 || ?def2 > ?upp1) . }                         (3)
```

Software-Ports

Operation – Same required/provided operation

```
SELECT ?m1 ?m2 ?p1 ?p2 WHERE {
    ?m1 :hasPort ?p1 . ?m2 :hasPort ?p2 . ?p1 :isConnectedTo ?p2 .
    ?p1 :hasType [ :hasOperationName ?o1 ; :hasFeatureDirection ?d1 ] .
    ?p2 :hasType [ :hasOperationName ?o2 ; :hasFeatureDirection ?d2 ] .
    FILTER ( ?d1 = ?d2 || ?o1 != ?o2 ) . }                                       (4)
```

Block/Module

Functionality – Fulfillment of same functionality

```
SELECT ?n WHERE {
    : Crane1 :hasFunctionality [ :hasName ?n ] .
    FILTER NOT EXISTS {
        : Switch1 :hasFunctionality [ :hasName ?n ] } . }                        (5)
```

Mandatory ports – Mandatory ports must be connected

```
SELECT ?m ?f ?p1 WHERE {
    ?m :hasFunctionality ?f . ?p1 :isMandatoryFor ?f .
    FILTER NOT EXISTS { ?p1 :isConnectedTo ?p2 } . }                             (6)
```

to the provided operation “Circular Movement”. If a module is connected to the crane via *S1*, it must be ensured that one of the connected modules requires and the other one provides the respective operation. Hence, the query (4) is applied ensuring that the operations (variables *o1* and *o2*) of the respective connected ports (variables *p1* and *p2*) refer to the same operation and that their feature directions (variables *d1* and *d2*) correspond to each other. In the application example, *S1* has the feature direction *Provided* and thus provides the operation *CircularMovement*. Hence, in order to be compatible, *S1* needs to be connected to a port that requires (feature direction *Required*) the respective operation *CircularMovement*.

In order to ensure that the system's functionality remains, if a module is exchanged, the new module needs to provide the same *functionality*. For the application example, the module *Crane1* is replaced by module *Switch1*; thus, it must be ensured, that *Switch1* fulfils the same functionality *f* being provided by *Crane1*. The SPARQL query (5) is therefore applied to retrieve the functionality provided by *Crane1* and to filter the results to identify the functionality not fulfilled by *Switch1*.

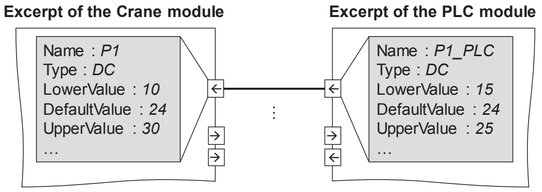

Fig. 8. E/E-connection of the Crane's port *P1* with the PLC's port *P1_PLC*.

In the example, both fulfil exactly the same functionality; hence, the query returns no result. Furthermore, it must be ensured that the ports that are *mandatory* in order to fulfil a specific functionality are connected to according ports of other system elements, cf. SPARQL query (6). In the application example, *P1* is mandatory for the functionality *Move WP Circularly* and, thus, must be connected to a respective port in order to provide the circular movement of a work piece.

5. Evaluation of the proposed approach

In order to evaluate the applicability of the approach, a change of the stack cylinder (for separating the work pieces) was conducted additionally to the example used in the preceding sections. Fig. 9 shows the old cylinder (left) and the new cylinder (right); their connections to respective ports of other modules are not shown. The old cylinder was made up by two 3/2-way valves, two end position sensors, and represented in the software by two software-blocks for extending and retracting the cylinder (Fig. 9 shows the resulting ports of the cylinder, the inner structure is regarded as black box). Thus, in order to fulfill the functionalities “Extend” and “Retract”, all ports are mandatory. The new cylinder uses one 4/3-way valve. Next to the software-blocks for extending and retracting the cylinder, it is also possible to extend the cylinder to a certain length and stop it in the respective position. The corresponding software port “Set Extension Length” is only mandatory if the cylinder shall be stopped in other positions than the end positions.

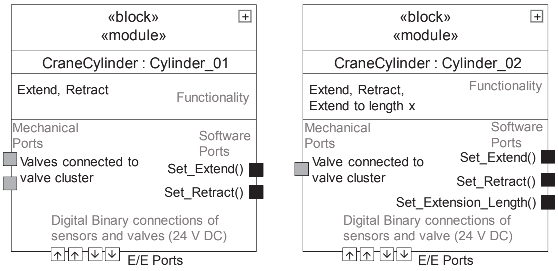

Fig. 9. SysML representation of old/ new cylinder with brief statement of port functionalities.

Similarly to the transformation in section 4, the SysML4*Mechatronics* models of the cylinders were transformed into the OWL ontology. Using the SPARQL queries (1)–(3), the connected ports' compatibility can be identified – all ports are connected correctly. Furthermore, as both cylinders provide the operations “Extend” and “Retract”, which are required by the PLC the cylinders are connected to, all operations required by the system are fulfilled and SPARQL query (4) identifies no compatibility issues. In addition, as the new cylinder fulfils an extended functionality, SPARQL query (5) identifies no problems regarding the functionality of both cylinders. As the valves were connected correctly (cf. previous queries), all mandatory ports needed for the module's functionality were connected properly, i.e. the valves required to perform the functionalities “Extend” and “Retract” are connected to respective ports. Hence, also SPARQL query (6) identifies, that an integration into the system is possible. This change analysis therefore showed that the new cylinder can be implemented into the system, although the working principle is different. However, if the enlarged cylinder functionality "Extend to length x" shall be used, it is necessary to adapt the other software blocks of the existing system.

The evaluation of the approach showed that it is applicable for lab-size demonstration models. However, the proposed approach deals with a pure structural compatibility analysis and, hence, is limited to a structural change influence analysis. In order to provide a more holistic analysis, behavioral aspects need to be integrated. By that, it could be automatically identified, whether the exchanged elements fulfil the same functionality as the original ones. Moreover, in order to make the approach applicable for complex manufacturing systems, the automatic validation of composed and more complex system elements must be provided. Nevertheless, using the proposed approach, influences of exchanged system elements on respective dependent elements can be analyzed and visualized. Thus, the approach has the potential to support engineers of different disciplines while designing complex manufacturing systems.

6. Conclusion and outlook

In this paper a SysML-based modeling approach and the adequate formal representation in an OWL ontology was shown, in order to enhance the possibility to analyze change influences. The combination of these two approaches appears very useful, as the SysML model offers the developers from different disciplines a comprehensive possibility to create an interdisciplinary system model with all required information, while the ontology can be utilized for the formal compatibility check. Thus, a first step towards providing (semi-)automatic compatibility checks for interdisciplinary models is provided. A major benefit of the used approach SysML4*Mechatronics* is the integration of the disciplines involved in the engineering process, e.g. mechanics, electrics/electronics, software, as in this way discipline-specific, as well as interdisciplinary dependencies between system components can be modeled, and thus, taken into account for the analysis of change influences. However, the integration of further aspects into the modeling approach, e.g. different variants and versions of system elements, will be investigated.

In this paper, the compatibility rules were defined beforehand and constitute general rules which have to be fulfilled by each exchanged system element. In future work, these rules will be extended to offer the possibility to define element-specific compatibility-rules (e.g. a maximum allowed energy demand for a module). In this way, also project-specific compatibility rules can be taken into account, enhancing the benefit of a model based change analysis even further. Furthermore, the purely syntactic description of operations will be extended towards a semantic description. Moreover, the proposed approach will be integrated into the existing modeling environment using automatic transformation between the SysML4*Mechatronics* model and the OWL ontology. By that, change influences can be analyzed in the formal model and after the reverse transformation visualized in the SysML model. Moreover, further research will be conducted towards enabling (semi-)automatically resolving incompatibilities by identifying appropriate elements from a central library. Together with experts from industry, the applicability of the approach for complex manufacturing systems will be evaluated.

Acknowledgements

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding parts of this work as part of the collaborative research centre 'Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products' (SFB768).

References

[1] Arafa A, ElMaraghy WH. Enterprise Strategic Flexibility. CIRP Conf Manuf Syst. 2012. pp. 537–542.

[2] Malak RC, Aurich JC. Software Tool for Planning and Analyzing Engineering Changes in Manufacturing Systems. CIRP Conf Intell Comput Manuf Eng. Elsevier B.V.; 2013. pp. 348–353.

[3] Chryssolouris G. Manufacturing Systems: Theory and Practice. 2nd ed. New York: Springer-Verlag; 2006.

[4] Bassi L, Secchi C, Bonfé M, Fantuzzi C. A SysML-Based Methodology for Manufacturing Machinery Modeling and Design. IEEE/ASME Trans Mechatronics 2011;16:1049–1062.

[5] Herzig SJI, States U, Reichwein A. A Conceptual Framework for Consistency Management in Model-Based Systems Engineering. ASME Int Des Eng Tech Conf Comput Inf Eng Conf. Washington DC, USA: 2011. pp. 1–11.

[6] Kernschmidt K, Vogel-Heuser B. An interdisciplinary SysML based modeling approach for analyzing change influences in production plants to support the engineering. IEEE Int Conf Autom Sci Eng. Madison, US-WI: 2013. pp. 1113–1118.

[7] Thramboulidis K. Overcoming Mechatronic Design Challenges: the 3+1 SysML-view Model. Comput Sci Technol Int J 2013;1:6–14.

[8] Shah AA, Kerzhner AA, Schaefer D, Paredis CJJ. Multi-view modeling to support embedded systems engineering in SysML. Graph Transform Model Eng. Berlin, Heidelberg: Springer-Verlag; 2010. pp. 580–601.

[9] Hundt L, Drath R, Lüder A, Peschke J. Seamless Automation Engineering with AutomationML. Int Conf Concurr Enterprising. 2008. pp. 685–692.

[10] Makris S, Alexopoulos K. AutomationML server – A prototype data management system for multi disciplinary production engineering. Procedia CIRP 2012;2:22–27.

[11] Legat C, Lamparter S, Vogel-Heuser B. Knowledge-Based Technologies for Future Factory Engineering and Control. Borangiu T et al., editors. Serv Orientat Holonic Multi Agent Manuf Robot. Berlin, Heidelberg: Springer-Verlag; 2013. pp. 355–374.

[12] Runde S, Fay A. Software Support for Building Automation Requirements Engineering – An Application of Semantic Web Technologies in Automation. IEEE Trans Ind Informatics 2011;7:723–730.

[13] Dibowski H, Kabitzsch K. Ontology-Based Device Descriptions and Device Repository for Building Automation Devices. EURASIP J Embed Syst 2011;1–17.

[14] Lastra J, Orozco O. Semantic Extension for Automation Objects. IEEE Int Conf Ind Informatics. IEEE; 2006. pp. 892–897.

[15] Lohse N, Hirani H, Ratchev S. Equipment ontology for modular reconfigurable assembly systems. Int J Flex Manuf Syst 2006;17:301–314.

[16] Abele L, Legat C, Grimm S, Andreas WM. Ontology-based Validation of Plant Models. IEEE Int Conf Ind Informatics. 2013. pp. 236–241.

[17] Strube M, Runde S, Figalist H, Fay A. Risk minimization in modernization projects of plant automation. IEEE Conf Emerg Technol Fact Autom. 2011. pp. 1–8.

[18] Bartelt C. Inconsistency Analysis at Integration of Evolving Domain Specific Models based on OWL. Int Conf Model Driven Eng Lang Syst. 2009. pp. 27–37.

[19] Hamani N, Dangoumau N, Craye E. Verification and validation of a SSM model dedicated to mode handling of flexible manufacturing systems. Comput Ind 2009;60:77–85.

[20] Legat C, Folmer J, Vogel-Heuser B. Evolution in Industrial Plant Automation: A Case Study. Annu Conf IEEE Ind Electron Soc. 2013.

[21] Institute of Automation and Information Systems. The Pick and Place Unit – Demonstrator for Evolution in Industrial Plant Automation, 2013. Available: http://www.ais.mw.tum.de/ppu/.