Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 93 (2020) 532–537

www.elsevier.com/locate/procedia

53rd CIRP Conference on Manufacturing Systems

A Graphical Toolkit for IEC 62264-2

Laurens Lang[a,b,c,*], Bernhard Wally[b,**], Christian Huemer[c], Radek Šindelář[b], Alexandra Mazak[b], Manuel Wimmer[b]

[a]*GoingMeta.io, Vienna*

[b]*CDL Model-Integrated Smart Manufacturing, Institute of Business Informatics – Software Engineering, JKU Linz, Altenberger Straße 69, 4040 Linz, Austria*

[c]*Business Informatics Group, Institute of Information Systems Engineering, TU Wien, Favoritenstraße 9–11/E194-03, 1040 Vienna, Austria*

Abstract

Among the plethora of industrial standards available in the context of smart manufacturing, one series of standards is consistently being mentioned for dealing with manufacturing operations management: IEC 62264. Its second part provides a conceptual model for the description of production systems and their capabilities, including runtime information such as concrete maintenance schedules or achieved production goals. In this work, we present a concrete graphical syntax and toolkit for the creation and presentation of IEC 62264-2 compliant models, using techniques from model-driven (software) engineering. We have evaluated our tool by conducting a user study for assessing its usability.

Keywords: IEC 62264; ISA-95; End User Development; Model-driven Engineering; Industry 4.0; Smart Manufacturing; Usability Engineering

1. Introduction

The current evolution step in manufacturing technology, Industry 4.0, has been accompanied by researchers with great enthusiasm [5]. It promises the digitization, automation and individualization of the complete manufacturing process, starting from product requirements and product design, over order processing and production to product delivery [7]. To materialize this vision, a variety of business partners that execute specific processes are required to work together ever closely, and potentially provide their capabilities as services. Two kinds of high-level system integrations have to be taken into account: (i) *vertical integration* for the integration within one partner, reaching from the business floor to the shop floor, and (ii) *horizontal integration* for the linking of systems among different parties by improving data exchange through evolved communication schemas and leveraging increased data quality [35].

Among the foundational aspects that may be required for realizing such interconnected, reactive manufacturing and supply chain systems, model-driven engineering (MDE) has been elevated to play a key role. MDE offers a huge palette of tools and techniques for the description and manipulation of formal models. Based on this, the creation of a *model-driven smart factory* seems reasonable—MDE promises to support the implementation of highly adaptable and reconfigurable systems, simplifying connectivity to other systems. This is to be achieved by a modular and semantic decoupling of diverse concerns, including mechanical, electrical, and control design as well as operational data from various levels of the automation hierarchy, such as enterprise resource planning (ERP) and manufacturing operations management (MOM) [2].

Nowadays, markets change rapidly and consumers demand constantly increasing product quality. The ability of different systems to exchange, understand, and utilize product, production, and business data relies heavily on information standards [17]. The usage of standards simplifies the process of adopting technologies for business owners.

However, not all standards are easily accessible, in that they can be applied without too much effort of learning the standard itself. Graphical modeling languages can help users to understand underlying abstract concepts more efficiently. In this work, we present a graphical concrete syntax for the specification of IEC 62264 models, a prominent standard in the field of digital manufacturing.

[*] Corresponding author. Tel.: +43-670 208 6108.
[**] Corresponding author. Tel.: +43-732-2468-4241.
E-mail addresses: laurens.lang@goingmeta.io (Laurens Lang)., bernhard.wally@jku.at (Bernhard Wally).

2. Related Work

Usability Engineering. In order to engineer user-oriented systems, there is a need for a systematic, empirical and consistent user-oriented development process model. Based on the famous *The Usability Engineering Lifecycle* we followed three main phases [20] : (i) *Requirements Analysis* The context analysis constists of a user, goal, task and constraint analysis. (ii) *User Interface Development:* First, the views of the task analysis were connected to a workflow. In the next steps, *Low-fidelity*, rough sketches of UIs have been sketched, discarded, discussed and refined. If they reached a certain degree of maturity, *high-fidelity mockups* were created in such a way that they look very similar to the end system. Recurring elements, design and interaction concepts were manifested in a styleguide in order to reach consistency and semiotic clarity. Only after these iterative steps, the implementation was executed. Future work will deal with even more process evolution techniques, usability hacks, which kickstart and possibilities to move the editor to the cloud.

Visual Notations. Visual notations play an enormous role in software engineering. Empirical studies confirm that the visual appearance of Reqirements Engineering notations significantly affects understanding [15]. Graphical diagrams are more effective than text in the goal-oriented requirements process, clarified by diagrammatic reasoning, which shows that the form of representations has an equal, if not greater, influence on human understanding and problem-solving performance as their content [15, 16]. Principles like *Semiotic Clarity*, *Perceptual Discriminability*, *Complexity Management* have been used to design cognitively effective visual notations and use the whole spectrum of the design space [26].

End User Development. Involving the end user by designing *software for change* leads to cost reduction. In the era of digitization, software applications will be developed, customized, and maintained by non-professional programmers. There are about 90 million estimated end users, who perform basic programming activities in American workplaces (165 million employed in total) [34]. As a consequence, the goal of human-computer interaction shifts from *easy of use* to *easy to develop*. For this purpose, environments have to be created which allow end users, who are familiar with basic computer functionalities and interactions to develop, modify and change systems. End-user development summarizes methods, techniques, and tools, which help end users who are neither skilled nor interested to create, adapt or evolve software or software artefacts [1].

Metadesign. From a more socio-cultural perspective this represents an empowerment of modelers. With the help of our tool, it is possible to create a functional IEC 62264 model without any further technical knowledge. Fischer et al. introduce *Metadesign*, which is an emerging conceptual framework aimed at defining and creating social and technical infrastructures, in which new forms of collaborative design can take place [6]. Fischer et al. determine, that these systems promote the transcendence of the individual mind (through collaboration), support sustained participation and enable the mutual adaptation and continuous evolution of users and systems [8].

Meta-Metamodels. Meta-Metamodels such as the Eclipse Modeling Framework (EMF) or the Meta Programming System (MPS) offer versatile tool platforms for the declaration and implementation of domain-specific languages and tools [14, 33]. With the Meta Object Facility (MOF) several approaches have been captured in a single standardized convention [28]. We are following the MOF definitions by relying on EMF's Eclipse-inherent implementation, Ecore. It provides a powerful ecosystem of tools that enable language designers to create modeling tools themselves.

Modeling in the Context of Manufacturing Systems. A good overview of standards that are relevant in the field of smart manufacturing is given in [17]. Among the plethora of standards that cover formal metamodels or domain-specific models are: IEC 62714 (AutomationML, [11]), IEC 62541 (OPC UA, [9]), and IEC 62264 (ISA-95, [10]). One kind of standard that is often left out in manufacturing landscapes is that of ERP systems. One standard that deals with the modeling of business entities, with a bias towards accounting theory is ISO/IEC 15944-4, better known as the Resource-Event-Agent (REA) business ontology [13].

IEC 62264 is specifically of interest because it is one of the three dominant standards depicted in the reference architecture model for Industry 4.0 [4]. It can be used to connect the MOM layer to the ERP layer, and it can be used to model MOM entities, functions, and activities. In this work we will develop a graphical concrete syntax for the second part of IEC 62264.

REA was first presented in [25] and has since then evolved into a powerful busines model language that even got standardized [13]. A graphical concrete syntax for REA has been developed in [22, 21]. A similar approach is aspired in this work, except that we are utilizing a different tool stack and are implementing the syntax for a different language.

The different standards and their domain-specific models and languages overlap in certain areas, which results in information duplication. In order to streamline this issue, mapping standards have been developed that mitigate some of the problems that may arise from that, such as data inconsistency. A strongly model-driven approach with its formalized descriptions can support the definition of clear interfaces and data exchange at design time and at runtime.

Fig. 1 summarizes how the noted standards are connected with each other: (i) OPC UA and AutomationML are mapped through a DIN SPEC [3], (ii) OPC UA and IEC 62264 are mapped through an OPC UA companion specification [29], (iii) IEC 62264 and B2MML are mapped "by design" [19], (iv) for the mapping between IEC 62264 and AutomationML there exists an official application recommendation [36], and (v) the mapping between IEC 62264 and REA has been drafted academically [23, 24, 38, 39, 37].

DECLARE. Process segment dependencies of IEC 62264 describe temporal and logical dependencies and restrictions between process segments. As they represent a declarative way of

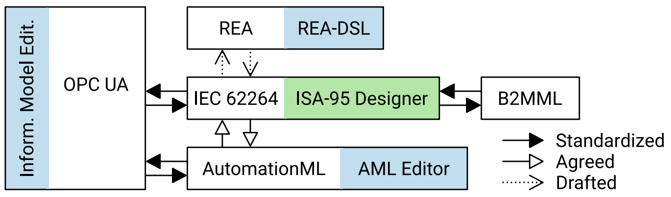

Fig. 1. Overview of strongly related standards around IEC 62264. Shaded boxes depict domain specific editors that are available. *IEC 62264 Designer* (green) is the editor described in this work.

specifying such process dependencies, it makes sense to use a declarative modeling paradigm in the visual concrete syntax. One constraint-based process modeling language for the development of declarative models describing loosely-structured processes is *DECLARE* [31]. Since the original DECLARE definition lacked a detailed description of temporal dependencies, a dialect of *DECLARE* arose: *timed DECLARE* [18]. It guarantees the correct execution of processes in terms of latency and deadlines and allows the monitoring of metrics of temporal constraints during runtime [18].

3. A Graphical Toolkit for IEC 62264-2

One of the main challenges in the process of designing a production system is the plethora of engineering disciplines that are involved and their corresponding individuals with their different personality traits, roles, skills and objective targets [30]. This challenge can be mitigated by improving communication among the different stakeholders, which in turn can be supported, e.g., by using and understanding a common language. IEC 62264 is a versatile standard that is able to encode information relevant for business (ERP), and for production (MOM). As such it seems suitable to play an integral role in the design of digital manufacturing systems. In this section we are presenting the visual concrete syntax of our graphical toolkit, the IEC 62264 Designer (*Designer* in short). Overall, we have developed 30 different views that allow reading and writing model information in a number of sub-models of the IEC 62264 standard. In this article, we are presenting six selected views thereof.

We will be using a running example throughout this section: the Industry 4.0 testbed deployed at the Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC) of the Czech Technical University (CTU) in Prague. It is an experimental, yet fully functional flexible production system used for teaching and research efforts alike. Details of this production system are being reveiled over the next sections while explaining the look of the *Designer*.

3.1. Basic Resources

In the first step, a visual notation for the basic resource models of IEC 62264 is defined. Due to space constraints, only *Equipment* is explained, *Personnel*, *Physical Assets*, and *Material* are left out.

Equipment. Fig. 2 shows the equipment hierarchy: purple rectangles represent pieces of equipment (hierarchically structured), the dark purple rectangles depict potentially attached thereto represent equipment classes. In this sense, CTU is an *Enterprise*, *CIIRC* is a subordinate *Site*, hosting on its Ground Floor (*Area*) the Testbed (*Production Line*). The testbed comprises a transport system, as well as several cells (*Work Cell*), which in turn each contain a table and a robot (*Units*).

The orange rectangle on Robot 1 represents the concrete physical asset that is currently installed and running in the production line. This is mapped via an "equipment-asset-mapping", as it is defined in IEC 62264.

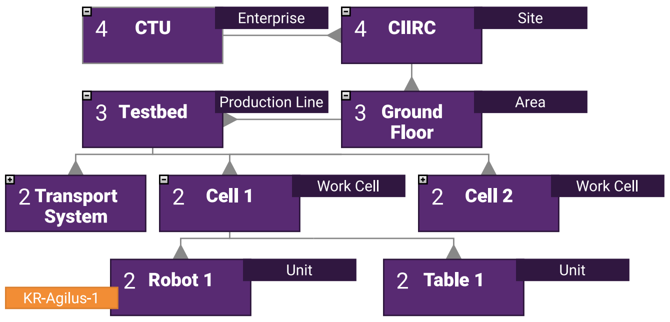

Fig. 2. Visual syntax for an IEC 62264 equipment hierarchy.

3.2. Process Segments

Figure 3 shows the process segment "Assemble" in detail. The cog-wheel icon in the top right corner of the main rectangle identifies this process segment as being of type *Production*. The process segment specifies a produced and a consumed material segment, both of material class "Truck Component". One equipment segment is specified in terms of "Table 1", and a physical asset segment in terms of a "KUKA Agilus" industrial robot. Here, no personnel segment is specified.

The small *plus sign icon* in the bottom center of the main rectangle symbolized that this process segment comprises child process segments. In this case a "Pick" and a "Place" process segment are defined underneath. Double-clicking on the process segment rectangle would reveal these process segments.

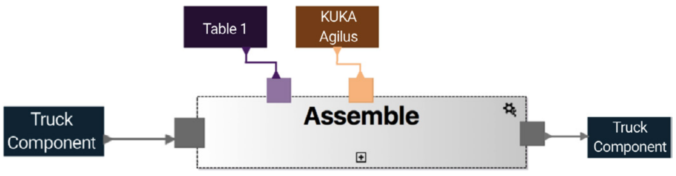

Fig. 3. Process segment "Assemble" with its resource specifications.

3.3. Operations Definitions

Product Definition. The products to be produced in our production system are toy trucks, assembled from chassis, cabin, and tank. In the product definition, which is in fact an operations definition of operations type *production*, more information is collected about a certain material definition, as it would be defined in a basic resource view. Fig. 4 depicts in the upper left corner a rendering of the product to be assembled, while the main part of this figure depicts the decomposition of the product definition: operations dependencies can be viewed under the "Dependencies" button, while the material bill (shown to the right) can be accessed through the "Material Bill" button.

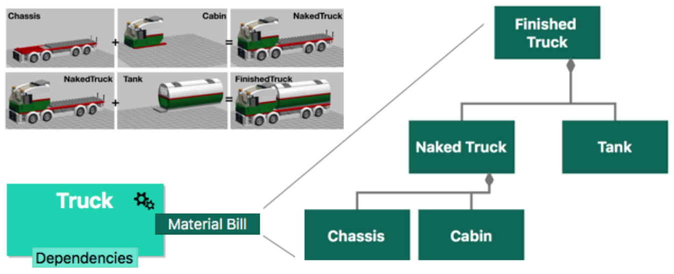

Fig. 4. Product definition (bottom left), material bill (right), and virtual product rendering (top left).

Operations Segments. In Fig. 5 an overview of the operations segments that make up an operations definition is given. This view supports concepts of BPMN [27]. The startsymbol symbolizes that "Place Chassis" is the first operations segment in this operations definition. After that, "Place Cabin" and "Place Tank" (in any order) can be executed. Afterwards, "Transport" and, finally, "QA" may be executed. The arrows between the operations segments depict material flow calculated from material specifications being produced and consumed.

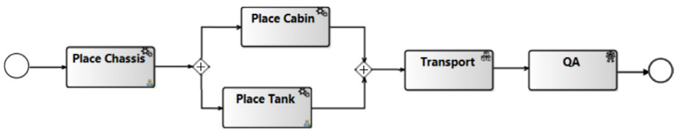

Fig. 5. Operations overview: shows the operations segments, which are connected through materials flow that are computed from material specifications.

Operations Segment Dependencies. Fig. 6 shows a selection of possible operations segments dependencies, based on the DECLARE syntax. Semantic overlap with the BPMN-like visualization is intended.

- "Place Chassis" is the start segment.
- "QA" must be executed at least one time.
- "QA" is the final segment.
- Once "Transport" has executed, "QA" must follow.
- "Place Cabin" can only be executed if "Place Chassis" was executed before.
- "Place Tank" can only be executed if "Place Chassis" was executed before.
- "Place Chassis" cannot follow "QA".

4. Evaluation

For our evaluation we have conducted a user study of our *Designer* in three different university departments from different universities, with a total of 14 participants. One university department had a background in software engineering, another one in business informatics, and a third one in control engineering. It was our goal to have a variety of experts in our test person pool: software engineerins for the assessment of the model-driven realization, business information experts for the view from ERP to MOM, and control/automation engineers for the view from MOM to ERP. Since each of the groups was too small for a dedicated assessment, we evaluated collectively.

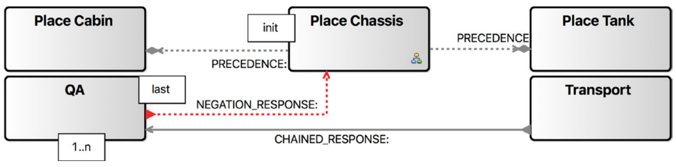

Fig. 6. Operations segment dependencies are defined using DECLARE syntax.

Our user study was a comparison of the *Designer* with the generic tree-based pseudo-graphical editor provided by EMF. Our hypothesis was that users that executed tasks with the *Designer* would (i) be faster and (ii) make less mistakes than those using the EMF editor. We prepared five tasks for this comparison (Tasks 3–7), and each user had to use the two tools alternatingly. There was no time restriction for task completion.

The study was executed as follows: First, the participants received a general introduction to IEC 62264 based on the official standard documents. Second, they were placed in front of prepared laptops equipped with the *Designer* and the EMF editor. Third, they could take ten minutes on their own to get accustomed to the *Designer* by playing around with it (Task 1; not part of the evaluation). Fourth, they had to paint a small production process with pen and paper (Task 2) in order to assess the visual notation naturally used by our participants for such a purpose. Fifth, they had to execute Tasks 3–7 with the *Designer* and the EMF editor alternatingly—half of the participants started with the *Designer*, the other half started with the EMF editor. Sixth, they filled out a questionnaire for qualitative assessment (user satisfaction) and demographic information.

Metrics. Usability can be defined as the perfect ratio of the relationship of tension between *Efficiency*, *Effectiveness* and *Satisfaction* [12]. During the study, participants had to model different aspects of a flexible production system. We have measured the time needed to complete the tasks, as well as the error rate (wrongly created elements and missing elements). These two characteristics address effectiveness and efficiency.

We have made use of the HEART framework [32], which focuses on user-centered metrics and defines a process for mapping product goals to metrics:

Happiness (H): subjective aspects, like satisfaction, visual appeal, likelihood to recommend, and perceived ease of use.

Engagement (E): the user's level of involvement with a product. The term is normally used to refer to behavioral proxies such as the frequency, intensity, or depth of interaction over some time period.

Adoption (A): how many users adopt the product.

Retention (R): how many users come back to the product.

Task Success (T): in our user study: the effectiveness (errors) and the efficiency (time).

Results. Table 1 shows the average numbers of errors for each task that were made with the EMF editor (left) and the *Designer* (right).

Table 1. Average number of errors that participants made within each task.

Task	EMF Editor	Designer
Task 3	1.14	0.50
Task 4	1.33	0.67
Task 5	1.00	0.33
Task 6	1.33	0.14
Task 7	0.86	0.00

Fig. 7 shows the amount of time that participants needed to perform each of the tasks. The top row displays results achieved with the EMF editor, the bottom row the ones obtained by using the *Designer*.

The participants had to complete a short qualitative questionnaire about their impressions on the tool. It was our goal to find out more about the perceived feelings with and the potential future use of our toolkit. Because, apart from the pure efficiency and effectiveness numbers it is also important to evaluate whether users are actually happy with the tool so that they would (i) *like* to use it and (ii) feel that it would be of help. Figure 8 shows the collected answers. For the first two questions about the users' happiness, a five-point Likert scale with respective answers ranging from "EMF editor" to "*Designer*" is used. The value in the center denotes a neutral opinion. The polar questions below surveyed the users' engagement, adoption, and retention.

Discussion. From the presented results, we can derive the following key insights:

- The quality of the models could be improved with the tool, since in all tasks less errors were made.
- The speed of reading models is significantly reduced with the help of *Designer*, which can be seen in the much faster executions of task 7.
- BPMN is the notation, which most users have intuitively used for modeling processes (captured in task 2). Using this kind of notation thus makes sense, for, e.g., displaying operations definitions.
- Using DECLARE for visualizing process and operations segment sependencies is an appropriate means for modeling restrictions. However, nearly all participants misunderstood a certain concept.
- Participants seemed to have liked the *Designer*. Users reported that it was more fun to use the designer than the generic tool and that they would adopt the tool, if they had to model IEC 62264 information.

However, we could not prove unequivocally the efficiency of the tool. The temporal numbers do not deliver a clear picture on which approach is better suited. One reason for this issue could be that the *Designer* is meant to be an expert tool. Learnability plays an enormous role for the overall usability, which could not be measured within this study. A longer-term study with a more realistic setup promises to unveil the engineering efficiency.

5. Conclusion

In this article, we have presented a visual toolkit that provides 30 different views for empowering end users to develop IEC 62264 compliant models. We have evaluated our implementation in a user study carried out in different facilities with varying user backgrounds and could infer that the usage of the tool increases the quality of the developed models and supports the users by re-using established visual languages such as BPMN and DECLARE.

While it is hard to generalize from a single study, we have learned the following lessons from our efforts: (i) user interface scalability to varying screen resolutions is a must, (ii) the re-use of known notations is beneficial, (iii) a visual notation may not always bee good for creating models, but is very efficient for reading them (if designed properly), (iv) a short introduction to the visual toolkit might have improved results significantly, and (v) re-use of a restricted set of interaction modalities throughout the toolkit fosters ease of use.

Unfortunately, we could not prove the efficiency of the tool in all the developed views, which, however, gives us a good backlog of improvements that we are going to implement in our next steps. Among the issues that we have discovered were (i) problems with varying display resolutions (the tested prototype was inadvertently optimized towards a specific resolution), (ii) tool usage misconceptions (that could be potentially learned quickly and improve efficieny for second- or longer-time users), and (iii) bugs in ther underlying tooling platform that led to system crashes. It was interesting to note that our tool was specifically well suited for the reading of information, especially when it was representing complex information, such as in the case of operations segment dependencies.

Besides bug fixes, future work will also deal with the supporting of additional concepts from IEC 62264, most notably resource relationship networks that are defined in part 4, as well as workflow specifications.

Acknowledgements

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the Nat. Found. for Research, Technology and Development is gratefully acknowledged.

References

[1] Barricelli, B.R., Cassano, F., Fogli, D., Piccinno, A., 2019. End-user development, end-user programming and end-user software engineering: A systematic mapping study. Journal of Systems and Software 149, 101–137.

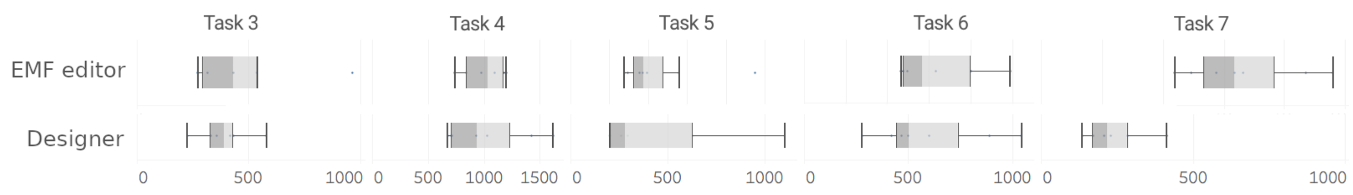

Fig. 7. Time required to execute the given tasks, in seconds. EMF editor (top), *Designer* (bottom).

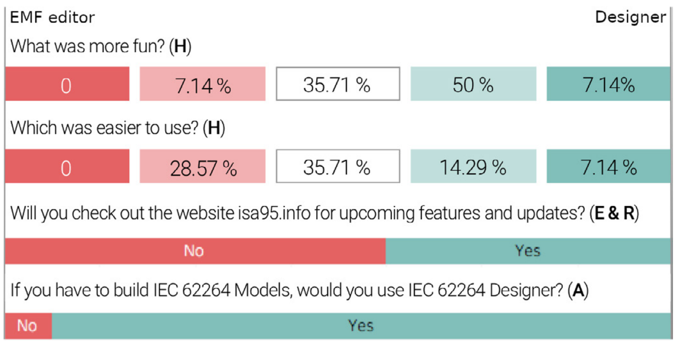

Fig. 8. Results from the subjective questionnaire.

[2] Cadavid, J., Alférez, M., Gérard, S., Tessier, P., 2015. Conceiving the model-driven smart factory, in: Proc. Int. Conf. Software and System Process, ACM. pp. 72–76.

[3] Deutsches Institut für Normung, 2016a. Combining OPC Unified Architecture and Automation Markup Language. DIN 16592:2016-12.

[4] Deutsches Institut für Normung, 2016b. Referenzarchitekturmodell industrie 4.0 (RAMI4.0). DIN 91345:2016-04.

[5] Drath, R., Koziolek, H., 2015. Industrie 4.0: Im Spannungsfeld zwischen dem Machbaren und Sinnvollen. atp edition – Automatisierungstechnische Praxis 57, 28–35.

[6] Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N., 2004. Meta-design: A manifesto for end-user development. Commun. ACM 47, 33–37.

[7] Gausemeier, J., Klocke, F., Dülme, C., Eckelt, D., Kabasci, P., Kohlhuber, M., Schön, N., Schröder, S., 2016. Industrie 4.0 – International Benchmark, Options for the Future and Recommendations for Manufacturing Research. Technical Report. Heinz Nixdorf Inst., Univ. Paderborn, Werkzeugmaschinenlabor WZL der Rheinisch-Westfälischen Techn. Hochschule Aachen.

[8] Giaccardi, E., Fischer, G., 2008. Creativity and evolution: A metadesign perspective. Digital Creativity - DIGIT CREAT 19, 19–32.

[9] International Electrotechnical Commission, 2010. OPC Unified Architecture – part 1: Overview and concepts. IEC 62541-1:2010.

[10] International Electrotechnical Commission, 2013. Enterprise-control system integration—part 1: Models and terminology. IEC 62264-1:2013.

[11] International Electrotechnical Commission, 2018. Engineering data exchange format for use in industrial automation systems engineering – automation markup language – part 1: Architecture and general requirements. IEC 62714-1:2018.

[12] International Organization for Standardization, 2006. Ergonomics of Human-system Interaction – Part 110: Dialogue Principles. ISO 9241-110:2006.

[13] International Organization for Standardization, International Electrotechnical Commission, 2007. Business transaction scenarios – accounting and economic ontology. ISO/IEC 15944-4:2007(E).

[14] Kern, H., Hummel, A., Kühne, S., 2011. Towards a comparative analysis of meta-metamodels, in: Proc. SPLASH'11 co-located WSs, pp. 7–12.

[15] L. Moody, D., Heymans, P., Matulevičius, R., 2010. Visual syntax does matter: Improving the cognitive effectiveness of the i* visual notation. Requirements Engineering 15, 141–175.

[16] Larkin, J.H., Simon, H.A., 1987. Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11, 65–100.

[17] Lu, Y., Morris, K., Frechette, S., 2016. Current standards landscape for smart manufacturing systems. Nat. Inst. Stand. Technol. .

[18] Maggi, F.M., 2014. Discovering metric temporal business constraints from event logs, in: Proc. Int. Conf. Business Inf. Research, pp. 261–275.

[19] Manufacturing Enterprise Solutions Association International, 2013. Business to manufacturing markup language.

[20] Mayhew, D.J., 1999. The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. 1 ed., Morgan Kaufmann.

[21] Mayrhofer, D., 2012. REA-DSL: Business Model Driven Data Engineering. PhD dissertation. TU Wien.

[22] Mayrhofer, D., Huemer, C., 2012. REA-DSL: Business model driven data-engineering, in: 14th IEEE Int. Conf. Comm. and Enterpr. Comp., pp. 9–16.

[23] Mazak, A., Huemer, C., 2015. From business functions to control functions: Transforming REA to ISA-95, in: Proc. 17th IEEE Conf. Business Informatics, pp. 33–42.

[24] Mazak, A., Huemer, C., 2015. HoVer: A modeling framework for horizontal and vertical integration, in: Proc. 13th IEEE Int. Conf. Industrial Informatics, pp. 1642–1647.

[25] McCarthy, W.E., 1982. The REA accounting model: A generalized framework for accounting systems in a shared data environment. The Accounting Review 57, 554–578.

[26] Moody, D., 2009. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering 35, 756–779.

[27] Object Management Group, 2013. Business Process Model and Notation (BPMN).

[28] Object Management Group, 2016. OMG Meta Object Facility (MOF) Core Specification. MOF 2.5.1.

[29] OPC Foundation, 2013. OPC Unified Architecture for ISA-95 common object model.

[30] Partsch, H., 1998. Requirements-Engineering Systematisch. Springer.

[31] Pesic, M., Schonenberg, H., van der Aalst, W.M.P., 2007. DECLARE: Full support for loosely-structured processes, in: Proc. 11th IEEE Int. Enterpr. Distributed Object Computing Conf., pp. 287–287.

[32] Rodden, K., Hutchinson, H., Fu, X., 2010. Measuring the user experience on a large scale: User-centered metrics for web applications, in: Proc CHI 2010.

[33] Savic, D., Rodrigues da Silva, A., Vlajic, S., Lazarevic, S., Antovic, I., Stanojevic, V., Milic, M., 2014. Preliminary experience using JetBrains MPS to implement a requirements specification language, in: Proc. 9th Int. Conf. Quality of Information and Communications Tech., pp. 134–137.

[34] Scaffidi, C., Shaw, M., Myers, B., 2005. Estimating the numbers of end users and end user programmers, in: IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 207–214.

[35] Schleipen, M., 2012. Adaptivität und semantische Interoperabilität von Manufacturing Execution Systemen (MES). PhD Thesis.

[36] Wally, B., 2018. Provisioning for MES and ERP. Application Recommendation. TU Wien. URL: https://www.automationml.org/.

[37] Wally, B., Huemer, C., Mazak, A., 2017a. Aligning business services with production services: The case of REA and ISA-95, in: Proc. of the 10th IEEE Int. Conf. Service Oriented Computing and Applications, pp. 9–17.

[38] Wally, B., Huemer, C., Mazak, A., 2017b. ISA-95 based task specification layer for REA in production environments, in: Proc. 11th Int. WS Value Modeling and Business Ontologies.

[39] Wally, B., Huemer, C., Mazak, A., 2017c. A view on model-driven vertical integration: Alignment of production facility models and business models, in: Proc. 13th IEEE Conf. Automation Science and Eng., pp. 1012–1018.