Evaluating Notations for Product-Service Modeling in 4EM: General Concept Modeling vs. Specific Language

Birger Lantow, Maria Dehne and Felix Holz

University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, Germany
{birger.lantow,maria.dehne,felix.holz2 }@uni-rostock.de

Abstract. The modeling of products and services has become a major area of interest in enterprise analysis. Increased complexity of products and the combination with services needs to be tackled. This paper compares two notations for product and service modeling that are suggested in conjunction with the 4EM Enterprise Modeling Method – a new Product-Service Model and the existing Concept Model. The comparison is based on Moody's principles for cognitively effective notations.

Keywords: 4EM, Conceptual Model, Visual Notation, Product Modeling, Enterprise Modeling.

1 Introduction

4EM or "For Enterprise Modeling" is a modeling method for enterprise modeling. It provides a procedure for the strategic problem oriented analysis of an enterprise and a notation for the modeling of certain problem domains. Although 4EM provides a modeling language the focus is not on notation. The central aspect of 4EM is the support of people in companies in modeling, finding improvement potentials and thus improving the company. A strong distinction is also made between the two perspectives of the modeler and the one who has to understand and interpret the models created. 4EM also provides a project-oriented and participative approach that should enable domain experts to create enterprise models with the help of a modeling expert. [1]

So far, 4EM defines six partial models for modeling the various areas and aspects of an enterprise. The modeling of products and services has become a major area of interest in enterprise analysis. Increased complexity of products and the combination with services needs to be tackled. Until now, none of the 4EM partial models provided a notation explicitly for modeling products and services. The Concept Model as one of the original six partial models can be used to freely define concepts like those that are needed for product or service modeling. As stated earlier, models need to be accessible to modelers as well as to those who interpret the models. When looking at understandability, readability, interpretability and further quality aspects of models and notations that have to be considered, not only the abstract syntax of a notation defining the concepts but also the symbols are relevant [2]. For this and other reasons, a new specific Product-Service Model has been developed as a partial model of 4EM. This can also be taken as an example of defining a Domain Specific Language (DSL). However, the focus of this investigation is not on the process of creating such a DSL but rather on the benefits that can be gained by a DSL compared to a general purpose approach like conceptual modeling. Still this is just an example, generalization of the findings will be future work. Section 2 describes the intended notation for product and service modeling and the possibilities of representing the required concepts using the 4EM Concept Model. The evaluation of the new notation is presented in Section 3. The discussion of Section 3 is based on Moody's principles for effective visual notations which will be introduced briefly. This is only a step in evaluating the proposed new notation for product and service modeling. Thus, the concluding Section 4 does not only summarize but also provides an outlook on further evaluation steps.

2 Modeling Products and Services with 4EM

This section describes how products and services and underlying structures can be modeled using 4EM. This is limited to a single model view. Relations to other partial models of the method like the Business Process Model are not considered here.
Section 2.1 describes the concepts and symbols defined for the new Product-Service Model. Section 2.2 then illustrates, how the used concepts can be modeled based the Concept Model notation. Thus, it will be possible to compare both possibilities of modeling products and services.

2.1 Product-Service Model

As already mentioned, the Product-Service Model is a proposed extension of the 4EM method. The Product-Service Model was specially developed to model products and services as well as their components. There are many notations for product and service modeling [3, 4, 5, 6]. It is a common base to model the composite structure of products and services and the resulting dependencies in it as well as market oriented features that carry value propositions. Modeling concepts of the new partial model have been selected based on this observation. The process of selecting these concepts will be described in a separate publication.

The most important concepts of the model are the product, the component and the feature. A product is an object of a company, which has a value for its customers and is offered by the company to its customers. Products can be further distinguished into services and goods (products in the narrower sense). The difference is that services are intangible and are co-created together with the customer [7] while the creation of goods does not need the customer involvement. Both can in turn be subdivided into any number of components or products/services of which they consist. In the terminology of the 4EM Product-Service Model there is the concept of an "Unspecific" product that cannot clearly classified, the "Service" concept, and the "Product" concept that refers to goods or products in the narrower sense. Features are properties that have a special value for the respective customer and which are realized by a product/service or one or more of its components.

The product service model also offers three different relation types: binary relations, generalization/specialization (ISA) relations, and aggregation (PartOF) relations.

Binary relations may have the following semantics. A Requires relation connects a feature to products/services, components and other features to show that these are required in order to implement the respective feature. Additionally, binary relations can be freely named and can thus carry semantics defined by the modeler. Furthermore, binary relations clarify the role of Product-Service Model elements in the following n-ary relations.

ISA relations are used to allow the construction of inheritance trees. The relation "A","B" ISA "C" expresses that product/service "A" and "B" are specializations of "C". This way variants of products, services or components can be modeled. "A" and "B" would be a special variant of "C". A specialization is total when all variants of the generic product or service are modeled. When the specialization is partial there may be variants that are not modeled.

PartOf relations are used to model the assembly or a product or service. The assembly shows the components that are needed to produce a product or provide a service. If components are themselves offered at the market by the company they can be modeled as products or services. Constraints of the assembly can be expressed by special PartOf relations. An AND relation indicates that all connected sub-components are required for the assembly. An OR relation indicates that connected sub-components are optional for the assembly. At last, a XOR relation indicates that only one of the connected sub-components can be part of the assembly at the same time. PartOf relations can be freely combined in order to express complex assembly constraints.
Fig. 1 shows the visual notation for Product-Service Model elements (from left to right, top-down): Component, Feature, PartOf-AND, PartOf-OR, PartOf-XOR, Total ISA, Partial ISA, Unspecified Product or Service, Product, Service.

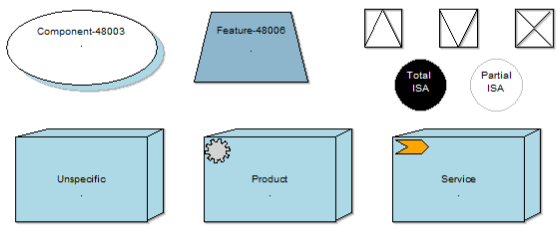

Fig. 1. Elements of the Product-Service Model

Fig. 2 shows a fictitious example of a Product-Service Model. The product "Mower Robot" consists of a "Chassis", a "Wheel Set", and a sensor for position control. Position control is done either using an "Inductive Sensor" or via "GPS + Galileo". Optionally, the "Mower Robot" can be equipped with a "LIDAR". The latter is required to provide the feature "Surveillance". There are two kinds of chassis – "Small Chassis" and “Big Chassis”. In order to deal with “Rough Terrain” the “Mower Robot” needs a “Big Chassis”.

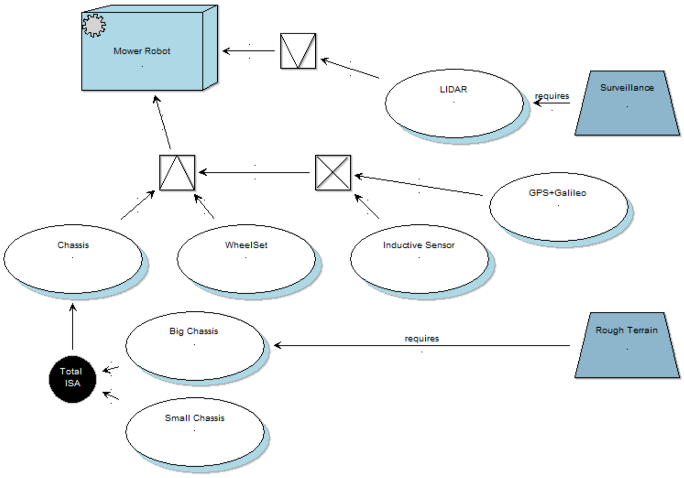

Fig. 2. Mower Robot - Example Product-Service Model

2.2 Concept Model

The Concept Model is mainly intended to describe concepts, terms and information objects used in the other 4EM-submodels in more detail in order to ensure a better understanding of them. This is particularly relevant if these concepts are used in several partial models in order to achieve a common, uniform understanding so that there are no misunderstandings or ambiguities.

Concepts can be used to model specific domains. Thus, if products and services are important for an Enterprise Modeling project, the Concept Model can be used to describe the structure of that domain and to define the semantics of products and services. It is an alternative to a specific Product-Service Model. Besides the definition of concepts, the model allows the addition of attributes and relations to the concepts such as binary relations, generalization/specialization (ISA) relations, and aggregation (PartOF) relations.

A Binary relation is a semantic relation between two concepts or within a concept. The modeler defines its semantics by naming it.

ISA relations are used to allow the construction of inheritance trees. The relation “A” ISA “B” expresses that concept “A” is a specialization of “B” and inherits “B’s” attributes. A specialization is total when all the instances of the generic type are members of one of the specified specializations. When the specialization is partial there may be instances of the generic concept that are not a member of any of the specializations. A PartOf relation is a form of semantic relation where the interrelated concepts are strongly and tightly coupled to each other. This could be used for example to model a product assembly. Similarly to the ISA relation, a total PartOf relation interrelates all partial concepts of a generic concept. In contrast, a partial PartOf does not need to be exhaustive. [1]

Fig. 3 shows the visual notation for Concept Model elements (from left to right, top-down): Concept, Attribute, Partial ISA, Total ISA, Partial PartOf, Total PartOf.
In order to model products and services using the Concept Model, the respective concepts have to be defined.

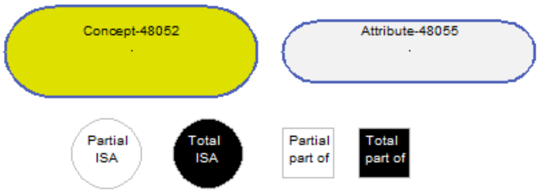

Fig. 3. Elements of the Concept Model

Fig. 4 shows the concepts corresponding to the notation defined in the previous section. Products, services, components, and features are modeled by a specialization of the core concepts. The Requires relation can be defined between a feature and any kind of assembly part. PartOf relations are predefined for the Concept Model. However, there is no differentiation between AND, OR, and XOR. It would be possible to create new assembly concepts for these. This would result in a change of Concept Model notation and thus a loss of generality of the Concept Model type and/or excess symbols for PartOf relations (see also criteria in next section). Besides the existing PartOf relations and its symbols, there would be additional symbols carrying the same semantics.

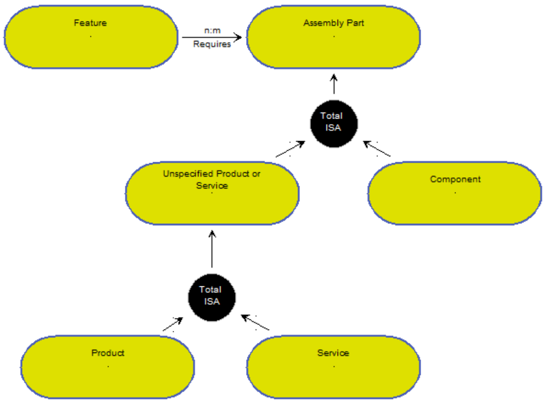

Fig. 4. Product-Service Concepts in Concept Model

Another possibility with some limitations would be to specify the roles of components that are interrelated by PartOf relations. Fig. 5 shows the mower robot example from the previous section modeled in Concept Model notation. Here, the optionality of

“LIDAR” is clarified by naming the binary relation between the “PartOf" and “LIDAR”. The exclusive alternative between the position sensors is expressed by specialization. However, this is only appropriate for semantically close concepts. Otherwise, alternatives would have to be modeled as abstract aggregate components that combine all alternatives.

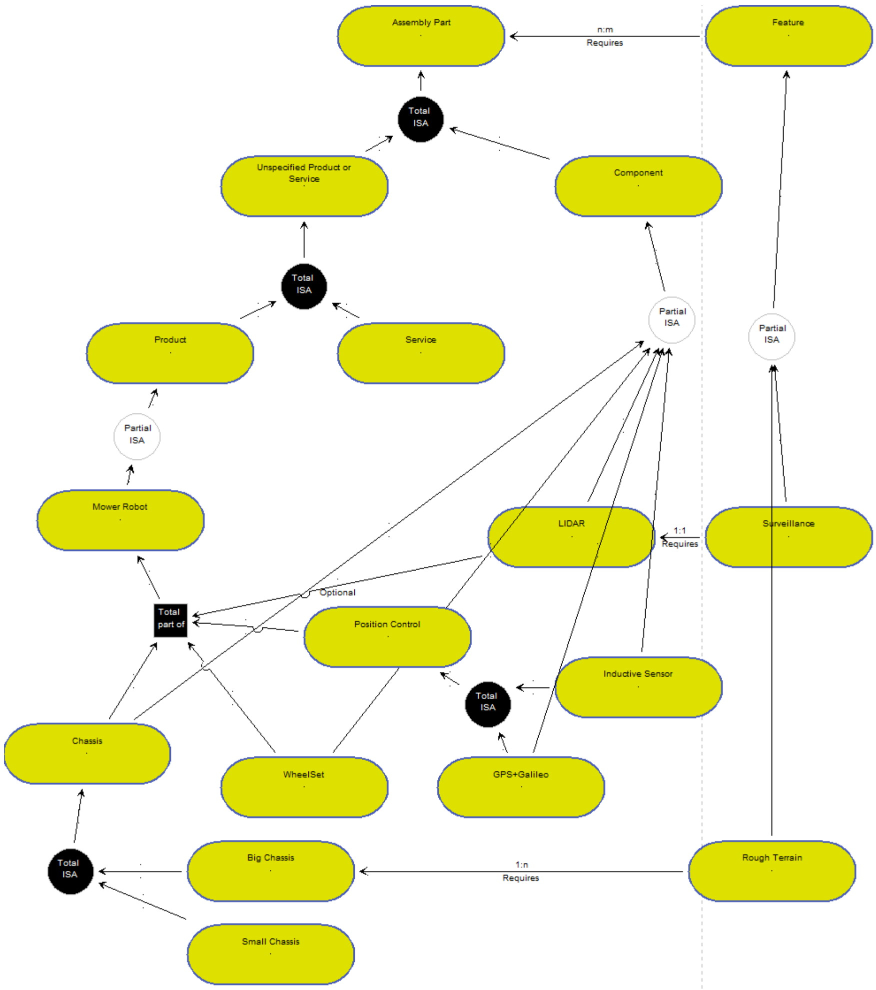

Fig. 5. Mower Robot - Example Concept Model showing Product Assembly and Features

3 Evaluation of the new Product-Service Model

In order to evaluate the new Product-Service Model in comparison to the use of the Concept Model, notation quality criteria are applied. Major work on the quality of visual notations is provided by Moody with his article “Physics of Notations” [8]. However, Moody mainly focuses on the comprehensibility of notations. Furthermore, he focuses on visual notation. There is more about a language than its visual notation. A distinction can be made between the symbols and the concepts behind them together with the abstract syntax for these concepts [2]. A more comprehensive approach to language quality that integrates Moody's work is the SEmiotic QUALity framework (SEQUAL) by Krogstie [9]. Referring to Section 2, there is no comparison of different concepts required and the focus should be on notation. Thus, an evaluation based on Moody's criteria is a relevant approach. Further experimental evaluation steps are planned for future. Section 3.1 describes the criteria proposed by Moody while the actual comparison is illustrated in Section 3.2.

3.1 Evaluation Criteria

Moody defines in [8] nine principles for the design of cognitively effective visual notations. These principles will be introduced briefly in the following as they are used for the comparison of product and service modeling using the domain specific Product-Service Model and using the general purpose Concept Model of 4EM.

Principle of Semiotic Clarity. This principle demands for a one-to-one correspondence between symbols and concepts of the notation. Moody defines four different possible deficits of visual notations with regard to Semiotic Clarity: (1) Symbol Redundancy, if multiple symbols represent the same concept (2) Symbol Overload, if a symbol represents multiple concepts (3) Symbol Excess, if there are symbols that do not correspond to a concept of the notation (4) Symbol Deficit, if there are notation concepts without a corresponding symbol. [8]

Principle of Perceptual Discriminability. The main idea of this principle is to make symbols clearly distinguishable from each other. Moody presents a number of suggestions in order to reach this goal, assuring Visual Distance of symbols by using a high number of visual variables like shape, color, size etc. to make them visually different. Shape is the most important visual variable here. Moody calls this fact Primacy of Shape. Further recommendations are the use of text to differentiate between symbols (Textual Differentiation), using unique values for at least one visual variable (Perceptual Popout), and to use more than one visual variable in order to make a difference between symbols (Redundant Coding). [8]

Principle of Semantic Transparency. Semantic Transparency describes extent to which the meaning of a symbol can be derived from its appearance. This can range from semantically immediate symbols where the meaning can be inferred without additional information over semantically opaque symbols where there is no link between appearance and meaning to semantically perverse symbols which imply a wrong meaning for the model user. The performance of a notation with regard to this principle depends on the model users and should thus be evaluated in experiments. However, Moody provides some general recommendations – the use of icons that depict real objects (Perceptual Resemblance) and special graphical relations (Semantically Transparent relations) such as intersections or trees. It can be checked whether these recommendations are implemented in a notation. [8]

Principle of Complexity Management. This principle demands for explicit mechanisms to deal with complexity. A simple measure for model complexity is the number of used elements. With increasing model size limits are reached regarding perception and cognition. Understandability of models suffers. Therefore, notations should provide mechanisms to reduce complexity. The main mechanisms to reach this goal are Modularization and Hierarchy (Levels of Abstraction). [8]

Principle of Cognitive Integration. There should be mechanisms to integrate information from different models. One mechanism would be Conceptual Integration. It provides an overview of the model and its sub-models by providing concepts on a high abstraction level that can be combined in order to relate the used sub-models. Perceptual Integration helps the model user with navigation in the model space. Since our goal is the comparison of partial model notations and not of the overall 4EM notation, this principle is not relevant at the current state of investigations. [8]

Principle of Visual Expressiveness. While Visual Distance (see above) considers the pairwise difference of concepts with regard to visual variables, Visual Expressiveness addresses the use of visual variables throughout the whole graphical notation. Hence, the question is which number of visual variables is used to express semantics and to distinguish between concepts (Information-carrying Variables) and which number of visual variables is not formally used (Free Variables). Moody defines a total of eight visual variables: Horizontal Position, Vertical Position, Size, Brightness, Color, Texture, Shape, and Orientation. The recommendation is to use as much Information-carrying Variables as possible. Consequently, there is a maximum of eight.

Principle of Dual Coding. Generally, text is not a good means to create a visual notation. However, there is a benefit of supporting visual notations by adding text. This can be done by Annotations and by Hybrid Symbols which combine text and graphical objects. [8]

Principle of Graphic Economy. This principle addresses the number of available graphical symbols for modeling. There is a recommended maximum of six symbols. An excess of symbols makes it difficult for the modeler to be aware of the symbols that can be used. Moody suggests three strategies for increasing Graphic Economy: (1) Reduce Semantic Complexity. Hence, the number of used concepts is reduced (2) Introduce Symbol Deficit (3) Increase Visual Expressiveness. [8]

Principle of Cognitive Fit. Here, different visual dialects are suggested for different tasks an audiences. The main assumption underlying this principle is that problem solving performance is influenced by the problem representation, task characteristics, and problem solver skills. Thus, a problem presentation should fit to the other two factors. This again requires involvement of model users for evaluation and is not considered at this stage of investigations. Furthermore, both compared notations do not provide dialects. Unless, you consider both as dialects of the same notation. [8]

There are also interdependencies between the formulated principles for notation design. For example, introducing a Symbol Deficit and thus reducing Semiotic Clarity fosters Graphic Economy. For the comparison of notations these interdependencies do not affect the evaluation but can be used to explain the characteristics of a notation.

3.2 Comparison of Product-Service Model and Concept Model

In the following the applicable principles by Moody are used to compare the new Product-Service Model with the Concept Model with regard to product and service modeling.

Principle of Semiotic Clarity. Looking into the four defined measures for Semiotic Clarity, there is no Symbol Redundancy in both notations. The Concept Model notation shows some deficits in the other measures (cf. Section 2.2). There is a Symbol Overload because the same shape is used for all concepts of product and service modeling. Depending on the way of decomposition modeling, there is a Symbol Deficit because there are no symbols for the special dependencies within assembly structures. Furthermore, there is a Symbol Excess considering the symbol for attributes (see Fig. 3). It is not used for product and service modeling. Considering the visualization of relations, there is a Symbol Overload in both model types because the same graphical representation is used for all binary relations. Additionally, Moody considers annotations as Symbol Excess. Both model types allow annotations as a core concept of 4EM. Thus, there would be a Symbol Excess in both of them. Overall, the Product-Service Model performs better than the Concept Model considering the principle of Semiotic Clarity. There are more issues of Symbol Overload, Symbol Excess, and Symbol Deficit in the Concept Model.

Principle of Perceptual Discriminability. Looking at the Visual Distance between any two concepts in the notations, the minimum distance is one for all concepts in the Concept Model because the only visual variable for distinction is a specialization relation to the respective core concept or text (cf. Section 2.2). The Primacy of shape is not adhered since all concepts have the same shape. The Product-Service Model (cf. Section 2.1) uses at least either shape or icons for the distinction of the concepts. Using also just one visual variable for differentiation considering “Unspecific”, “Service”, and “Product”, it performs better. Furthermore, the Product-Service Model uses Visual Popouts for components and features. N-ary inheritance relations are symbolized the same way in both models as well as binary relations. For assembly structures, the Product-Service Model uses shape as a visual discriminator while the Concept Model uses color to distinguish between Partial and Total PartOf. However, adding missing semantics to assembly structures in the Concept Model uses the same mechanisms that are used for the concepts like products and services. Textual Differentiation is possible in both models with regard to the concept. Redundant Coding is not further discussed. It is used wherever the visual distance is greater than one. For example for the distinction between concepts and assembly relations. Overall, the Product-Service Model performs better that the Concept Model regarding Perceptual Discriminability.

Principle of Semantic Transparency. As described in Section 3.1, an evaluation of this principle should involve experiments. First assumption can be made based on the use of icons and special relations (see Section 3.1). The Product-Service Model uses icons for “Product” and “Service”. Though not referencing directly to real world objects, the used icons are commonly used to refer to concepts of enterprise planning. Both models use the same general relation types. Overall, there is a slight indication that Semantic Transparency is better in the Product-Service Model.

Principle of Complexity Management. Modularization is not supported by any of the two models. Hierarchies however can be modeled using n-ary relations in the form of inheritance trees or assemblies. These are available in both models. Overall, none of the notations can evaluated better than the other with regard to Complexity Management.

Principle of Cognitive Integration. As discussed in Section 3.1 this principle is not used for evaluation in this investigation.

Principle of Visual Expressiveness. The Concept Model uses three Information-carrying Variables: Shape, Color, and Size. The Product-Service Model uses Shape, Color, Brightness and Size. Brightness is used because the features have a darker blue than the other concepts. There would be one more Information-carrying Variable in this notation. Besides the Visual Variables themselves also their coding range should be considered. Thus, how many different shapes, colors etc. are used? Considering Shape, there are nine different shapes in the Product-Service Model (including the icons) and three different shapes in the Concept Model. With regard to color there are five (white, black, grey, orange, blue including the icons) in the Product-Service Model and four (white, black, blue, yellow, excluding the unused attribute concept) in the Concept Model. Differentiation in size is applied in both models between n-ary relations and the other concepts. Overall, the Product-Service Model performs better in visual Expressiveness. However, there might be some bias by including/excluding certain model elements. Hence, the general tendency is obvious.

Principle of Dual Coding. Both models combine text and symbols for concepts and relations. Thus, there is no difference with regard to this principle.

Principle of Graphic Economy. Graphic Economy is evaluated by defining a maximum threshold for the number of graphical symbols. According to Moody this threshold is six. However, six applies only if the symbols are coded using only one visual variable. The Concept Model uses five different symbols (excluding attributes). The Product-Service Model uses ten symbols. However, looking at the maximum number of symbols that differ in only one visual variable it is three in the Product-Service Model (AND-OR-XOR and Unspecific-Service-Product). For the Concept Model it is five (Unspecific-Product-Service-Feature-Component, see Fig. 4). Actually, there is no variation in the visual variables defined by Moody. Thus, the threshold of six symbols is not exceeded by any of the two models.

Principle of Cognitive Fit. This principle is not applied in this investigation (see Section 3.1).

Summarizing the discussion, the Product-Service Model is expected to have a better cognitive effectiveness compared to the Concept Model. The Product-Service Model supersedes in the Principles of Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, and Visual Expressiveness while it is not worse in comparison regarding The other principles. While interpretation of the principles might differ in detail, the general tendency is clear.

4 Summary and Outlook

Based on the theoretical comparison of the new Product-Service Model to the Concept Model with regard to product and service modeling using Moody's principles, it can be concluded that the new model will perform better in terms of cognitive effectiveness. Thus, a better understandability and interpretability can be expected. However, theoretically clear circumstances may result in not so clear practical consequences as shown for example in [10]. Further evaluation is required in both directions. First, it needs to be proved that the selected concepts for product and service modeling are appropriate for practical problems and tasks. Furthermore, understandability and interpretability etc. should be evaluated in experiments in order to back the theoretical assumptions. We have also the effect of different roles, skills and tasks of the model users. This refers to Moody's principle of Cognitive Fit. In consequence, experiments should for example explicitly address the quality of the new model with regard to different tasks performed with the model e.g. the model creation and the model analysis. Such experiments have already been performed using an extension of the existing AdoXX-based 4EM modeling toolkit and are in the evaluation process right now.

References

1. Sandkuhl, K., Stirna, J., Persson, A., & Wißotzki, M.: Enterprise modeling Tackling Business Challenges with the 4EM Method. Springer, heidelberg (2014).
2. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: E-Commerce and Web Technologies. pp. 182–182. Springer, Berlin, Heidelberg (2002).
3. Riebisch, M.: Towards a more precise definition of feature models, 64–76 (2003).
4. Trevisan, L., Brissaud, D.: Engineering models to support product–service system integrated design. CIRP Journal of Manufacturing Science and Technology, 15:3 – 18 (2016).
5. Weber, C., Werner, H. et al.: Implizite und explizite Bestandteile des Produktmodells und ihre Bedeutung für die Entwicklung von CAx-Systemen . In: DFX 1999: Proceedings of the 10th Symposium on Design for Manufacturing, Schnaittach/Erlangen, Germany, 14.-15.10. 1999, pp. 1–6. (1999).
6. Pezzotta, G., Pirola, F., Rondini, A., Pinto, R., Ouertani, M.: Towards a methodology to engineer industrial product-service system – Evidence from power and automation industry. CIRP Journal of Manufacturing Science and Technology, 15:19 – 32 (2016).
7. Zeithaml, V.A., Parasuraman, A. & Berry, L.L.: Problems and strategies in services marketing. Journal of marketing, Vol 49.2, 33-46 (1985).
8. Moody, D.L.: The "Physics" of Notations: Towards a Scientific Basis for Con-structing Visual Notations in Software Engineering. IEEE Trans. Softw. Eng. 35, 23 (2009).
9. Krogstie, J.: Model-Based Development and Evolution of Information Systems: A Quality Approach. Springer, London (2012).
10. Lantow, B.: Level of Detail and Understandability of Enterprise Models–Better Understandability through Higher Complexity?. In: International Conference on Business Informatics Research (pp. 45-56). Springer, Cham (2014).