Practical experiences with multi-level modeling using FMML^x: A hierarchy of domain-specific modeling languages in support of life-cycle assessment

Monika Kaczmarek-Heß[1], Mario Nolte[1], Andreas Fritsch[2], and Stefanie Betz[3]

[1] University of Duisburg-Essen, Essen, Germany
monika.kaczmarek|mario.nolte@uni-due.de
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
andreas.fritsch@kit.edu
[3] Furtwangen University, Furtwangen, Germany
besi@hs-furtwangen.de

Abstract. In this paper we report on practical challenges we faced while designing a hierarchy of Domain-Specific Modeling Languages (DSMLs) spanning through a reference DSML, an industry-specific DSML and finally, an enterprise-specific DSML (ESML) supporting a life-cycle analysis of products. The mentioned hierarchy of DSMLs has been designed using a multi-level modeling approach FMML^x (Flexible Meta Modeling and Execution Language) and a supporting tool XModeler. Based on the faced challenges, we formulate a set of postulates regarding the further development of the field in order to provide better support for practical applications.

1 Introduction

In the discourse of Sustainable Development (SD) companies are requested to reduce unintended social and ecological impacts resulting from their activities. For this purpose Life-Cycle Assessment (LCA) has been developed and established in organizational practice over the last decades [14]. It allows companies to collect and manage information about potential environmental and social impacts of products along their life-cycle (e.g., information about emissions during production, usage and disposal of products or services). This information can be used for discursive decision making in support of SD.

In order to consolidate already existing procedures for gathering and processing information and to allow for a comparison of results, standards like ISO 14040 were developed containing generic terms and procedures [20, 14]. The on-going development of LCA showed however, that not only further concepts were needed (e.g., midpoint [10]), but also that the results of LCA studies remained complex and not easy to interpret and communicate [19].

Inspired by the latter, in our earlier work we show how conceptual modeling can be used to support LCA by developing a modeling language *TracyML* [11]

and a modeling method *ImpactM* [24] that both support the collection and presentation of information related to social or ecological impacts. Both languages are based on a *conventional language paradigm* where modeling languages were defined by meta models that can be used to develop models one language-level below (i.e., M_1). Although both languages showed their applicability in different scenarios, the integration and on-going developments revealed the need to adapt and extend them to address the information needs of different domains and incorporate relevant knowledge. While this need could be satisfied by the development of various Domain-Specific Modeling Languages (DSMLs), we deemed this approach as inefficient, since multiple DSMLs would have to be maintained and also relevant knowledge would have to be respecified from scratch.

To overcome this conflict between reuse and productivity of DSMLs [15], i.e., a conflict between generic and specific concepts, in our current work we propose a hierarchy of DSMLs that contains domain-specific primitives and allows to customize them further into a family of specialized languages: (1) a generic definition of a reference DSML (rDSML) supporting LCA in general, (2) further definition of the modeling language by refining the generic LCA concepts to meet domain/industry specific needs (DSMLs), and finally, (3) usage of a language in the context of an enterprise as Enterprise-Specific Modeling Language (ESML). As, among others, such a hierarchy spans multiple classification levels, requires deep instantiation and treating classes as objects, therefore, we have designed it using a multi-level language architecture [6, 15].

Although the resulting multi-level model is applicable in various scenarios, the application of the multi-level modeling approach to such a complex domain as LCA resulted in numerous practical challenges. As until now the practical applications of multi-level modeling to domains of such complexity are still rather scarce [1], the main goal of this paper is to present our observations and formulate a set of recommendations, which should point to possible directions of development of the multi-level field. Our observations result from the application of the Flexible Meta Modeling and Execution Language ($FMML^x$) together with a supporting tool XModeler [15]. Therefore, they are not representative for all existing multi-level modeling approaches and supporting tools.

The paper is structured as follows. First, basic concepts connected with life-cycle assessment and challenges in this field are introduced. Then, we present our main goal and vision to show how we intend to use multi-level modeling in support of LCA. Next, we present selected observations and recommendations. The paper concludes with final remarks.

2 LCA And Related Concepts

In the realm of SD the tool of LCA has become important for companies to identify and document information about potential ecological and social impacts that are related to different states of a product or service (e.g., extraction of raw materials, usage, disposal). For this purpose a *product system* is specified by outlining activities that are considered as relevant as well as activities or materials

Table 1: Selected concepts proposed in ISO 14040 for LCA [20, pp. 7-14]

Term	Definition
product system	"collection of unit processes with elementary and product flows, performing one or more defined functions, and which models the life cycle of a product" [p. 11]
sys. boundary	"set of criteria specifying which unit processes are part of a product system" [p. 11]
functional unit	"quantified performance of a product system for use as a reference unit" [p. 10]
impact category	"class representing environmental issues of concern to which life cycle inventory analysis results may be assigned" [p. 13]
category endpoint	"attribute or aspect of natural environment, human health, or resources, identifying an environmental issue giving cause for concern" [p. 12]
raw material	"primary or secondary material that is used to produce a product" [p. 9]

that are excluded from the assessment (e.g., materials below a certain weight). It allows to identify, qualify and quantify several ecological or social impacts that can be traced back to activities within the product system. In order to lay out basic requirements and to make results of different ecological assessments comparable the standard ISO 14040 was introduced. Aside of common language concepts as presented in Tab. 1 the standard provides guidelines that describe the application of these concepts. Even though these general steps and concepts allow for applying the standard in several domains, the definitions, as presented in Tab. 1, already indicate that refinements in accordance to the context and purpose of each study are necessary. In consequence, several terms suggested by ISO 14040 are characterized by different attributes and for various purposes. For example the term *impact category* comprises concepts like Global Warming Potential (GWP) or Resource Depletion (RD) of specific resources. While the former, i.e., GWP, is of relevance for answering ecological questions and requires attributes like conversion factors that allow for the expression of Carbon Dioxide Equivalents, the latter one, i.e., RD, is of relevance for economic procurement questions [9] and requires attributes like the amount of known resources in the earth crust, the anthropogenic stock ingrained in products used (e.g., cars) or the renewal rate of biotic resources, as well as information of local stocks of resources. It becomes clear that the concepts are specified in a generic way that leaves freedom for a user to apply the standard to different domains by defining purpose and attributes in accordance to his or her needs. This generic nature of the standard resulted in a huge amount of literature, methods and software tools that reference concepts proposed by ISO 14040.

Considering the above, there are a few challenges that LCA is still facing, cf. [19, 13, 9]. They are connected with (1) a further need for standardization while accounting for different application scenarios; (2) a constant need to integrate new developments; (3) acquisition and quality of data required for the needs of assessment; and finally, (4) ensuring comparability of performed studies.

3 Main Goals And The Designed Multi-Level Model

As already mentioned, to address the above mentioned challenges, our goal is to offer a hierarchy of DSMLs spanning through a reference DSML that includes concepts for conducting LCA, which are refined to specific industrial domains in

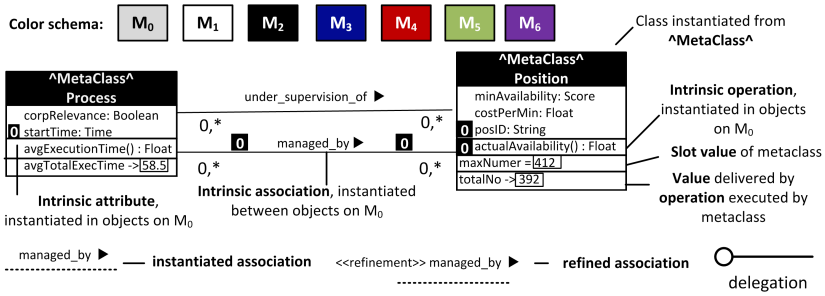

Fig. 1: Concrete Syntax of FMML$_x$ [15]

form of DSMLs with an increasing level of specificity, up to certain enterprises as ESMLs. Multi-level modeling was identified as an instrument of choice to realize the above-goal, as the application of conventional two-level modeling paradigm, although *technically* possible, would impose important limitations. These limitations would hinder us from delivering a satisfactory language specification, i.e., a solution without workarounds, overloaded levels, model redundancy and accidental complexity [7, 15, 12]. As in multi-level modeling there is no strict division between language specification and application [6], all languages (generic, regional, local ones) are represented within a single model (a multi-level model) and thus, users can access all classification levels they are interested in (cf. Fig. 2). In addition, thanks to the usage of a multi-level approach, we can also benefit from such features as relaxed type/instance dichotomy or deep instantiation.

We have selected FMML^x for our design due to the fact that it offers a common representation of model and code [15] and comes with an integrated language execution engine offered by a supporting tool XModeler [15]. This feature allows us, among others, to equip models with behavior and to provide support for computational analysis, which is necessary to perform ecological or social impact assessments as mentioned in Section 2. Whereas the detailed description of FMML^x may be found in [15, 17], Fig. 1 presents its concrete syntax. Apart from the "conventional" modeling constructs such as classes, attributes, operations and relationships, it is possible to deffer an instantiation of all modeling constructs by assigning them so called level of intrinsicness, which tells at what level a property will be instantiated. Intrinsicness differs from well-known potency regarding its interpretation: while potency uses the relative number of levels (e.g., two levels below), the level of intrinsicness tells us at what absolute classification level (e.g., M_0) the given property will be instantiated. Although there is an on-going work on the extensions to the FMML^x's abstract syntax to increase its expressiveness, for the time being no additional concepts like the ones offered by deep modeling [2] (e.g., mutability) are supported.

The presented excerpt from a multi-level model (Fig. 2) shows a hierarchy of professional terminology used during the LCA analysis. While the rDSML provides generic ISO concepts, all of the refined DSMLs provide semantically rich concepts with a wide range of properties (cf. a hierarchy from a Raw Material,

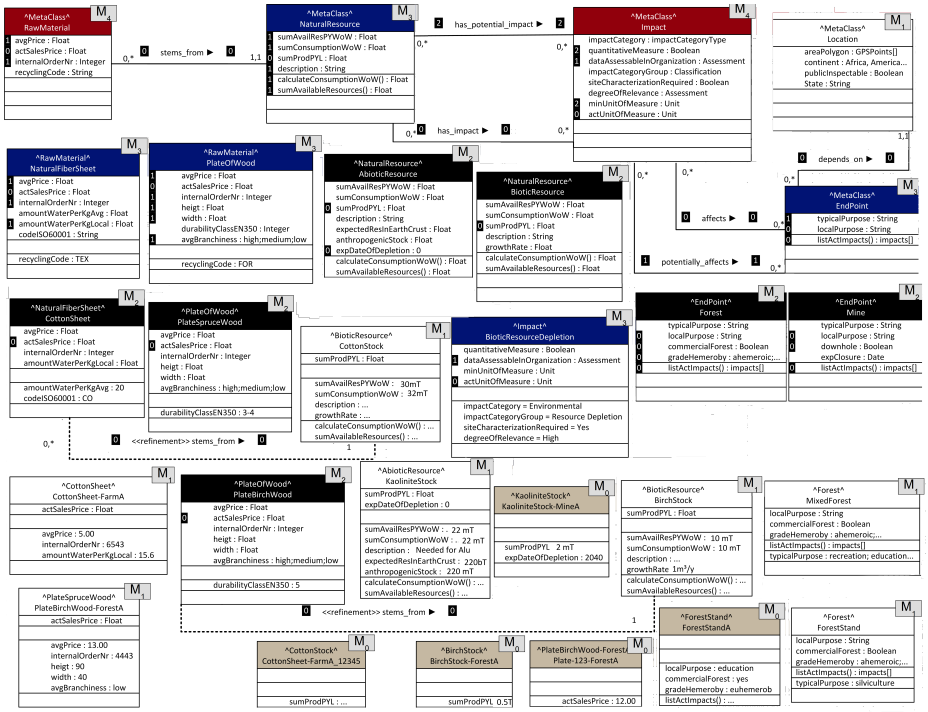

Fig. 2: An excerpt from the LCA multi-level model

Natural Fiber Sheet, to different types of Cotton Sheet and relevant instances). Thanks to the intrinsicness, we may state what is relevant as soon as we know it and defer its instantiation to some lower level. This supports integration and comparability of different methods. By offering concepts on different classification levels we support both productivity and reuse at the same time (e.g., we offer both an abstract concept *Resource* as well as a set of its specific types and instances ready to be used). Moving along the hierarchy we instantiate and specialize concepts at the same time. So on the one hand, we inherit and extend the definition of characterizing properties. On the other hand, thanks to relaxed type/instance dichotomy, we assign a state to classes, and thus, e.g., state what is the recycling code for a Plate of Wood. This way, the modeled concepts do also store information for the needs of assessment, which supports its productivity. Users of the language, even if they are not experts in LCA, can benefit from the incorporated knowledge. It encompasses, among others, a set of impacts, their indicators, assessment methods, requirements regarding the data to be used, as well as, if possible, the data itself that is needed for the needs of calculation. As in FMML^{x} a class is an object [15], operations can be not only specified for classes, but also executed on them (e.g., calculateConsumptionWoW()). Finally, the proposed hierarchy allows to conduct the analysis on the local and global

level. This means that a language user can apply one of the more specialized DSMLs to conduct the analysis of interest and then aggregate ("bottom-up") the results by moving up along the hierarchy. It is also possible to "drill down" in the model to individual localized impacts, that are relevant to specific stakeholders.

Please note that to (1) allow for application of different concepts from different *impact assessment methods* in tandem, (2) allow users to access all classification levels they are interested in, as well as (3) ensure the comparability of achieved results, all of the DSMLs in the hierarchy are integrated, through, among others, the refinement of concepts from the rDSML (vertical integration), and also through the definition of aligning horizontal relationships. Accounting for all of the above makes the model quite complex with multiple relationships, their refinements and multiple multi-level constraints.

4 Practical Observations

On the one hand, multi-level modeling frees modelers from differentiating between language specification and application, however, on the other hand, it empowers language users to be language designers as well [16]. The creation of multi-level models requires numerous changes in the mindset of modelers and language designers. Firstly, they need to think not only in two but in multiple layers. They need to be able to assign classification levels to different constructs and, in case of FMML^{x}, define operations. They also need to better manage the complexity: as all classification levels are integrated in one single model and also additional constructs may be used, multi-level models are usually more complex and semantically richer than 'conventional ones' [21]. Subsequently, we discuss the practical experiences we gathered when designing our multi-level model. Due to space restrictions only selected observations are reported on.

1. Lack of guidance and heuristics. As we have been well equipped with a domain knowledge on LCA and had a number of different scenarios at our disposal, the identification of relevant concepts and their properties was relatively easy. Nevertheless, assigning the concepts to the adequate classification level turned out to be very challenging. The initially assigned classification levels or assigned levels of intrinsicness needed to be adjusted a number of times. As no heuristics allowing to judge on the quality of a multi-level model are available, therefore, when faced with different alternative conceptualizations, only scenarios as well as our own experience in conceptual modeling were supporting our decision-making process. **Recommendation:** This challenge is present not only when working with FMML^{x}, but according to our knowledge is inherent to all multi-level modeling approaches. Indeed, a lack of guidelines and heuristics for designing a multi-level model and assess its quality has been already noticed by the multi-level modeling community [1]. When it comes to the quality of multi-level models, we consider the work of [8] as a starting point however, more specific operationalization is required.

2. A lack of satisfactory support for the initial phase of model creation. Considering that we needed to revisit the assigned levels of classifications as well as intrinsicness a number of times, the multi-level model was changing quite often during the initial phase of its creation. In addition, although most of the existing multi-level tools support the top-down creation process of multi-level models (cf. the constructive approach [5]), in our case a bottom-up approach (cf. exploratory approach [5]) together with a mixed-approach was much more efficient and effective. Therefore, designing the multi-level model straight in XModeler turned out to be not possible due to not yet implemented change propagation algorithms [25] and a lack of support for exploratory modeling. A sheet of paper did not help much either, as the model turned out to be too big and too complex to draw it. **Recommendation:** Although different styles of modeling (e.g., constructive and exploratory ones) are acknowledged, they are supported only by few existing tools, e.g., [5]. This limits the initial, creative phase of multi-level model development. Also during this phase a support for not-yet known aspects should be provided. In addition, it should be possible to document the decision-making process in the form of notes or comments, and to *assign defined properties to different application scenarios* that a multi-level model is to support. This could be used later on to generate, e.g., different views on created models (cf. Observation 7). To the best of our knowledge such a set of features is not available in any of the existing multi-level modeling tools. When it comes to the mentioned change propagation algorithms, they should also allow a user (with appropriate authorization) to modify the model, which is already in-use. The change propagation algorithms should be *transparent* and *interactive* in the sense that a tool should always ask users what they exactly want to change and ask how they want the change to be accounted for within the model.

3. Dealing with unbalanced hierarchies and contingent classifications. In FMML^{x}, similarly to most of existing multi-level approaches, the amount of levels in a hierarchy needs to be kept in balance. Our work indicates however, that there is the need to relax this constraint since classification levels of some classes are contingent depending on the branch that is followed. An example can be found in the class *EndPoint*, which is assigned in the excerpt of the multi-level model to language level M_3. Using the assignment of *potential impacts* as determination criteria for further concepts, the branches of *Forest* and *Mine* indicate that different language levels are needed: While in the case of forest potential impacts like occupational accidents can be assigned directly to the concept of a forest (and instantiated), the branch for mines needs further language levels since potential impacts can be assigned to different types of mines. E.g., potential impacts like gas explosions are related to underground mining, but have no relevance for surface mining which can however, be related to potential impacts like land use. **Recommendation:** There is a clear need for constructs that would allow to deal with unbalanced hierarchies and free modelers from creating 'artificial levels' to make the hierarchies even. This could be done by allowing for a contingent class or by introducing a leap potency (cf. [22]). A specific solution however, depends on the approach and its architecture.

4. Insufficient expressiveness and problems in defining multi-level constraints. Although in comparison to traditional modeling approaches, we could use many more constructs, we missed in FMML^x a possibility to express: (1) a contingent assignment of intrinsicness levels: quite often there was a need to define a range instead, i.e., the earliest level and latest level of classification when a certain value needs to be instantiated; (2) pointing which instantiated properties are representative also for further instances and down to which level. Although we thought at first that mutability [2] will be also required in our case, taking into account that (from the domain perspective) there is a semantic difference between attributes obtaining values (or changing values) at different levels, it was more natural to define different attributes that would relate to different classification levels. While defining refinements (i.e., constraints) on the relationships was relatively easy due to the built-in functionality of XModeler, we faced problems in expressing other constraints using XOCL [15], e.g., that the value of some property defined on higher levels restricts the possible values of some other property on lower levels. Although our lack of experience is partly to be blamed for it, we would welcome additional wizards and more interactive way to define the multi-level constraints of interest. **Recommendation:** There is still a need to reflect on the semantics of already existing constructs and keep the possibility to enrich the expressiveness of properties. Beyond any doubts applicability and necessity of each extension should be carefully considered so that additional constructs would not lead to an increased complexity of a created model. Regarding formulating multi-level constraints, we would find it reasonable to define in XModeler (and in other multi-level modeling tools) a set of templates or a wizard that would facilitate definition (and interpretation) of multi-level statements.

5. Unsatisfactory support for concrete syntax design. It would be unrealistic to expect that the prospective users of our hierarchy would be dealing with the excerpt of the multi-level model shown in Fig. 2. Here providing an adequate concrete syntax – adjusted on each level of a hierarchy to different industries and enterprises – is crucial. Unfortunately, the possibilities offered by XModeler to define a concrete syntax for multi-level models are still rather limited and a topic of on-going research, cf. [18]. Therefore, although notation plays a crucial role for the usefulness of any modeling language, for now only abstract syntax for our multi-level model is available. **Recommendation:** Already existing ways to define a concrete syntax for multi-level models, such as the one implemented in Melanee [4], should be analyzed in order to find the best of breed approach. In addition, a discussion of experiences in designing notation for multi-level DSMLs should be started. Resulting guidelines can help novice users and should consider human processing capabilities and lessons learned from the design of concrete syntax in the conventional modeling, e.g., [23].

6. Inadequate mechanisms for complexity management. While multi-level modeling does indeed help avoid *accidental complexity*, the resulting models in XModeler can get very complex very quickly (necessarily so, since they are intended to span several domains). However, human cognitive processing capa-

bilities are limited. While this has been discussed extensively for language users [23], one can safely assume that the same is true for language designers. It is an objective of multi-level modeling to provide flexibility ("power to the users"), yet this flexibility tends to get lost in the face of overwhelming complexity of the resulting models. Although XModeler offers numerous possibilities to hide different elements (e.g., operations, slot values), we missed more advanced mechanisms allowing to expand/collapse some hierarchies or showing only selected classification levels. **Recommendation:** There is a need to provide additional mechanisms for complexity management. These should be available within the provided meta-modeling tool, but could also be specified for the meta modeling language (e.g., the mentioned possibility to assign some properties to some specific application scenario, or to some perspectives/roles).

7. A support for defining and generating multiple views required. What we missed was the possibility to define and generate multiple views not only on the model itself, but also on the specific concepts. Different stakeholders with different perspectives/roles assigned need different views to enhance the comprehension of the complex model. **Recommendation:** Equipping existing tools with a possibility to generate different views on the created multi-level models seems to be a must. The possibilities should span multiple criteria: user role, application scenario, vertical and horizontal abstractions, etc. The existing work, e.g., [3] could be used as a starting point. It would be also reasonable to investigate possibilities to generate different abstractions, e.g., next to a static abstraction also a dynamic abstraction in a form of a process view.

5 Conclusions

Despite its undisputed benefits, multi-level modeling has still not reached the expected maturity and its application in practical scenarios does not come without challenges. In our opinion, the most important aspect is to provide, next to the required tool support, also a procedural support in form of process models, guidelines and heuristics, which would support inexperienced modelers and language designers in a transition from a two-level to multi-level way of thinking.

References

1. Almeida, J.P.A., Frank, U., Kuehne, T.: Multi-level modelling (dagstuhl seminar 17492) **7**(12), 18–49 (2018), in: Journal Dagstuhl Reports, ISSN 2192-5283,
2. Atkinson, C., Kennel, B., Gutheil, M.: A flexible infrastructure for multilevel language engineering. IEEE TSE **35**, 742–755 (04 2009)
3. Atkinson, C., Tunjic, C.: A deep view-point language for projective modeling. In: 2017 IEEE 21st EDOC. pp. 133–142 (Oct 2017)
4. Atkinson, C., Gerbig, R.: Aspect-oriented concrete syntax definition for deep modeling languages. In: Atkinson, C., Grossmann, G., Kühne, T., de Lara, J. (eds.) Proc. of 2nd Inter. Workshop on Multi-Level Modelling (MULTI). pp. 13–22 (2015)
5. Atkinson, C., Kennel, B., Goss, B.: Supporting constructive and exploratory modes of modeling in multi-level ontologies. In: ISWC 2011. pp. 1–15 (2011)

6. Atkinson, C., Kühne, T.: The essence of multilevel metamodeling. In: Proc. of the 4th Int. Conf. on The Unified Modeling Language, Modeling Languages, Concepts, and Tools. pp. 19–33. Springer, London (2001)
7. Atkinson, C., Kühne, T.: Reducing accidental complexity in domain models. SoSyM **7**(3), 345–359 (2008)
8. Atkinson, C., Kühne, T.: On evaluating multi-level modeling. In: Burgueño, L., et al. (eds.) Proc. of the 4th Inter. Workshop on Multi-Level Modeling (MULTI). pp. 274–277. CEUR-WS.org (2017)
9. Bach, V., Berger, M., Forin, S., Finkbeiner, M.: Comprehensive approach for evaluating different resource types–case study of abiotic and biotic resource use assessment methodologies. Ecological Indicators **87**, 314–322 (2018)
10. Bare, J.C., Hofstetter, P., Pennington, D.W., De Haes, H.A.U.: Midpoints versus endpoints: the sacrifices and benefits. The Int. Journal of LCA **5**(6), 319 (2000)
11. Betz, S., Fritsch, A., Oberweis, A.: Tracyml-a modeling language for social impacts of product life cycles. In: Cabanillas, C., Espana, A., Farshidi, S. (eds.) Proceedings of the ER Forum 2017 and the ER 2017 Demo track (2017)
12. De Lara, J., Guerra, E., Cuadrado, J.S.: When and how to use multilevel modelling. ACM Trans. Softw. Eng. Methodol. **24**(2), 12:1–12:46 (Dec 2014)
13. Finkbeiner, M., et al.: Challenges in Life Cycle Assessment: An Overview of Current Gaps and Research Needs. pp. 207–258. Springer, Dordrecht (2014)
14. Finkbeiner, M., Inaba, A., Tan, R., Christiansen, K., Klüppel, H.J.: The new international standards for life cycle assessment: ISO 14040 and ISO 14044. The international journal of life cycle assessment **11**(2), 80–85 (2006)
15. Frank, U.: Multilevel modeling - toward a new paradigm of conceptual modeling and information systems design. BISE **6**(6), 319–337 (2014)
16. Frank, U.: Power-modelling: Toward a more versatile approach to creating and using conceptual models. In: Proc. of 4th Int. Symp. on BMDS. pp. 9–19 (2014)
17. Frank, U.: Designing models and systems to support it management: A case for multilevel modeling. In: Atkinson, C., Grossmann, G., Clark, T. (eds.) MULTI@MoDELS. pp. 3–24. ceur-ws.org (2016)
18. Gulden, J.: An example application of a multi-level concrete syntax specification with copy-and-complete semantics. In: Burgueño, L., et al. (eds.) Proc. of the 4th Inter. Workshop on Multi-Level Modellng (MULTI) (2017)
19. Hellweg, S., Milà i Canals, L.: Emerging approaches, challenges and opportunities in life cycle assessment. Science **344**(6188), 1109–13 (2014)
20. ISO: DIN EN ISO 14044 - Umweltmanagement - Ökobilanz - Anforderungen und Anleitungen (2006)
21. Kaczmarek-Heß, M., de Kinderen, S.: A Multilevel Model of IT Platforms for the Needs of Enterprise IT Landscape Analyses. BISE **5** (2017)
22. de Lara, J., Guerra, E., Cobos, R., Llorena, J.M.: Extending deep meta-modelling for practical model-driven engineering. The Computer Journal **57**(1), 36–58 (2014)
23. Moody, D.: The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE TSE **35**(6), 756–779 (Nov 2009)
24. Nolte, M., Kaczmarek-Heß, M.: Product life-cycle assessment in the realm of enterprise modeling. In: The Practice of Enterprise Modeling. pp. 187–202 (2017)
25. Toepel, D., Benner, B.: Maintenance of multi-level models - an analysis of elementary change operations. In: Burgueño, L., et al. (eds.) Proc. of the 4th Inter. Workshop on Multi-Level Modeling (MULTI) (2017)