A Hierarchy of DSMLs in Support of Product Life-Cycle Assessment

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

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

Abstract. To support understanding and analysis of sustainability related aspects in organizations (e.g., via an assessment of a product's life-cycle from the cradle to the grave), various instruments, among others, in the field of conceptual modeling, have been proposed. Although existing tools and languages are, to some extent, indeed supporting the product Life-Cycle Assessment (LCA), our investigations show that a hierarchy of Domain-Specific Modeling Languages (DSMLs) is needed to satisfy advanced requirements. In this paper, as an innovation for the field of LCA, we propose an application of multi-level language architecture to design a hierarchy of DSMLs encompassing concepts for LCAs that can be detailed to specific industrial domains and local needs of enterprises. This enables a new generation of instruments allowing users to use and refine concepts, corresponding to their specific needs.

Keywords: LCA, sustainable development, multi-level modeling.

1 Introduction

To support the sustainable development (SD) of organizations, the awareness about ecological and social impacts of their products, potentially leading to unintended changes in the environment, needs to be increased. To increase that awareness, relevant information on potential impacts caused by all activities related to the production, usage and disposal of products, needs to be collected and used in decision-making processes [1, p. 226]. In this context *Life-Cycle Assessment* (LCA) has been established and used over the last decades to collect such information in a systematic manner [2]. Standardized by norms like ISO 14040 [3], LCA provides generic concepts and instructions that have been refined into several different assessment methods containing more specific concepts, which address the information needs within different industrial domains [4]. To support the application of LCA assessment methods various tools are provided, which produce complex results that are not easy to interpret and communicate, cf. [5]. In addition, currently existing LCA tools do not provide satisfactory support either [6,7], as results are not always transparent and traceable [8].

14th International Conference on Wirtschaftsinformatik,
February 24-27, 2019, Siegen, Germany

To mitigate these challenges our earlier work shows how conceptual modeling can be used as an instrument to collect, structure, aggregate, and present data about potential ecological and social impacts of products along their entire life-cycle. For this, we proposed a modeling language *TracyML* [9] and a modeling method *ImpactM* [10]. Due to contestedness of the idea of SD [11] both methods do not provide a solution on their own, but allow to document relevant information required for the needs of the discursive decision-making like assumptions and information about system boundaries allowing for the comparison of results of different assessments.

Both languages are implemented in a *conventional language paradigm*, where the language specification is defined using a meta model that can be used to develop models one language-level below (i.e., on M_1). As both languages are based on ISO, therefore, for the sake of reuse, they are kept rather generic. As a result, although showing their applicability in different scenarios, those languages exhibit a similar deficiency as ISO 14040. Namely, to satisfy the specific information requirements of industrial domains or enterprises, first, a substantial effort needs to be invested to define and/or adjust the required concepts during the language use. Although it would be possible to propose a variety of Domain-Specific Modeling Languages (DSMLs) and various diagram types in LCA tools to avoid a need for such an adaptation, this approach would result in a threat to efficiency. Indeed, not only multiple DSMLs and diagram types would have to be developed and maintained, but also relevant information, e.g., on typical impacts related to domain-specific resources, would have to be provided during the language use, which results in considerable time and cost expenditures.

To overcome these challenges the goal of our research is to propose a hierarchy of DSMLs spanning through: (1) a *reference Domain-Specific Modeling Language* (rDSML) [12, p. 321], which includes generic concepts for LCA, that can be refined to (2) *specific industrial domains*, up to (3) languages with a high level of specificity for certain enterprises as *Enterprise-Specific Modeling Languages* (ESMLs), cf. [13]. Our proposal addresses advanced requirements, as there is a need to provide software support for different user groups and purposes. On the one hand, researchers need generic and flexible tools, which allow for modelling of standard and diverging scenarios. On the other hand, for industry users efficiency and effectiveness is most important. They need pre-specified and easy-to-use software that can be easily parameterized [14]. At the same time, a major challenge in LCA and weakness of available LCA software is ensuring comparability and compatibility of user models [6]. Within the concept presented in this paper, the industry- and enterprise-specific modeling languages form the basis for efficient and easy-to-use software tools that cater for the specific requirements of enterprises and industries. Hereby, the underlying integrated hierarchy provides the conceptual foundation for the specific modeling languages, and ensures the comparability and compatibility of user models.

This contribution follows the design-oriented research paradigm [15]. The resulting IT artifact (i.e., the hierarchy of DSMLs developed in an iterative manner) aims at providing a benefit to organizations by supporting LCA and addressing the above-mentioned challenges. To create the targeted hierarchy, we follow the language design method proposed by [16], which provides a macro-process and corresponding roles, as well as a set of guidelines. In this paper we focus on three selected outcomes, i.e., clarification of goals and scope of the solution, identified requirements, and a resulting hierarchy of DSMLs. We also present a realistic scenario to illustrate our vision.

The paper is structured as follows. First, a short overview on general ideas of product LCA is provided. Then, an exemplary scenario follows, which is used to explain the vision of our research. Next, requirements towards a language architecture are shortly discussed. Then, we present the hierarchy of DSMLs in form of a multi-level model.

2 (Product) Life-Cycle Assessment

LCA studies aim at identifying all relevant impacts through-out the life-cycle of a product. Thereby, the *product system* is defined as the life-cycle containing different states of a product, from extraction of the necessary raw materials, via production and assembly of its components to usage and final disposal or recycling [3]. The ISO 14040 standard lays out basic requirements to avoid biased studies and inappropriate claims [17]. Thus, it provides a common language and guidelines how to apply it. The four main steps for conducting an LCA study comprise [3]: (1) goal and scope definition, (2) life-cycle inventory analysis, (3) life-cycle impact assessment, and (4) interpretation. During the *goal and scope definition*, in particular the system boundary is to be defined. Depending on information needs and practical constraints, one may, e.g., choose to address the whole life-cycle from resource extraction to disposal (“cradle to grave”), or just the assembly of some parts (“gate to gate”) and define cut-off criteria (e.g., excluding material below a specific weight). The second phase, the *life-cycle inventory analysis*, is where the data collection and definition of indicators takes place. Impacts are modeled in the third phase, *life-cycle impact assessment*. This is typically done using the concepts *impact category* and *category endpoint*, cf. Tab. 1. Note that these are very broad concepts and, as *different assessment methods* are available, cf. [18], their usage differs significantly. In the following, we use the term *impact* to describe an effect that may happen on a global (e.g., climate change), regional (e.g., smog, eutrophication) or local level (e.g., acidification of water), and can be traced back to a product system. Correspondingly, an *endpoint* may represent an entity that is affected by an impact (e.g., freshwater supply in a lake). Finally, in the last step, the assessment results are interpreted and communicated.

Table 1. Selected concepts proposed in ISO 14040 for LCA [3, pp. 7-14]

Term	*Definition*
product	“any goods or service”
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”
sys. boundary	“set of criteria specifying which unit processes are part of a product system”
funct. unit	“quantified perform. of a product system for use as a reference unit”
impact category	“class representing environmental issues of concern to which life cycle inventory analysis results may be assigned”
category endpoint	“attribute or aspect of natural environment, human health, or resources, identifying an environmental issue giving cause for concern”

Due to the applicability of the standard to various domains and contexts, the *concepts are specified on a general level* and the standard itself offers freedom to users in defining the functional unit, system boundary, data sources etc. As a result, a large amount of literature, standardization processes [19] and databases exists, which reference the ISO-concepts, e.g. [4]. Additionally, there are efforts to apply LCA to the social dimension of sustainability [20], and advance the method to provide an integrated ecological, social and economic perspective [21].

3 Motivating Scenario

Consider two companies intending to improve their sustainable development: Company T producing T-Shirts out of natural fiber, and Company S producing shelves composed out of wood and metal. In particular, those companies wish to reduce their ecological and social impact, while ensuring on-going and long-term economic success. The comprehensiveness of the LCA-approach as the "gold standard" of sustainability assessment has already convinced them to follow a life-cycle perspective on their products. They expect that this will help them identify all relevant impacts and avoid problem shifting between them.

However, once our companies start the analysis, it becomes unclear to them (as in-depth LCA expertise within companies is missing), what potential impacts would be relevant for the analysis in question. Likewise, both companies face difficulties with decisions to be made, e.g., what phases actually to include in the assessment. Thus, both companies are challenged with a need to reconstruct their domain knowledge and information needs using the generic concepts of the ISO-standard. As a result, due to practical constraints like data availability [19], and also different information interests, the resulting models created by companies are on various levels of abstraction and use data on various levels of detail.

Next, the *availability of resources* for raw materials on the local and world-wide level needs to be considered. While the growth-rate of different kinds of natural fiber is of interest for Company T, Company S has interest in the growth-rate of different kinds of wood. Because the shelves contain also metal parts, Company S is also interested in the availability of metal stocks (e.g., Bauxite as ore for aluminum), which can be expressed through the expected date of depletion of specific mines on a local level, or through resources expected in the earth crusts and anthropogenic stocks (e.g., used in cars) on a global level. Regarding the *raw materials* both companies share the interest in economic data, like sales and average prices, and the recycling code that might help at a later point in time determine the potentials for recycling raw materials.

Coming to discuss relevant potential and actual social and ecological impacts that can be traced back to raw materials, it becomes clear that a huge variety of impacts and related endpoints can be calculated by different assessment methods and related indicators. Indeed, regard, e.g., social impacts: while for the assessment of wood-based production systems impacts related to health & safety, employment or equal opportunities of forestry workers might be relevant, for fiber-based products impacts like toxic emissions related to the dying of cotton is of importance. Here it can be differentiated between effects from inhalation, which can get chronic or carcinogenic, and effects stemming from ingestion. Furthermore, because Company T intends to avoid the consumption of natural fiber as a raw material having a high need for water consumption with respect to its origin, this company defines the availability of water as an *endpoint.* Therefore, it requires information about the average amount of water for the raw material of natural fiber in general, as well as the amount of water for natural fiber consumed in a specific farm. Also, the soil of a farm is in this context important and treated as an endpoint, which is characterized by its fertility and erosion. In the long-term, Company T aims to lower the rate of farms supplying natural fiber, which have too high water consumption or a high grade of erosion. In turn, Company S intends to make use only of wood that stems from a forest management targeting at the production of wood and not from forests managed for other purposes, e.g., conservation, recreation. This information needs to be modeled as endpoints. In addition, since this company also intends to provide financial support for non-production forests that are untouched by human influence, the grade of hemeroby (e.g., ahemerobic for non-influenced forests) should be included in models too.

Finally, it is also important for both companies to have specific quality-related information on raw material used. Here, it is assumed that *branchiness* and the *durability class* in accordance to DIN EN 350-2 [22] are considered as relevant for the raw material of wooden products for Company S. In turn, Company T is interested in codes provided in the DIN 60001-1 [23] for textile materials, as well as in *elasticity* of different kinds of fiber, and in *tensile strength* assessing the capacity of fiber or other materials to elongate without tearing apart.

Table 2. Aspects of interest and related LCA concepts - overview

	Company T	*Company S*
Product	T-Shirts	Shelves
Raw material	Natural fiber (avg./sales price; recycling code)	Wood (avg. and sales price; recycling code); metal (avg. and sales price; recycling code)
Endpoint	Soil of farms (location; fertility; erosion)	Forest stands (location; commercial forest); mines (location; downhole)
Impact	Resource depletion natural fiber (kind; renewal rate; consump. worldw.; stock worldw.); virtual water	Resource depletion wood (kind of fiber; renewal rate; consumption worldw); resource depletion metals (kind; consumption worldw.; sum known resources); pot. social and environ. impacts
Resource	Stocks of different kinds of natural fiber, e.g., cotton, linen (global and local)	Stocks different kinds of wood, e.g. birch, spruce (global and local); stocks of different kinds of metal, e.g. aluminum
Level of detail	Typical / average information; specific stocks of resources	Typical/avg. inform.; resources specific stocks; individual product dependent information for certificates demanded by customers
Quality interests	Elasticity; tensile strength, Code DIN 60001-1	Branchiness (high, average, low); durability class ('1'=very durable, '5'=non-durable)

Tab. 2 gives a structured overview of the scenario. In the first three lines concepts from ISO 14040 are assigned to the specific interests of both companies. The fourth row presents different kinds of resources. Terms in the brackets should give an impression on possible exemplary attributes, values or further concepts that are necessary to support the impact assessment as sketched above. The two last lines indicate individual quality-related interests and levels of details that should be satisfied by conceptual models that support this scenario. Please note that only the ISO concepts themselves are readily available and all other ones need to be modeled or reconstructed, which significantly hampers the productivity of the analysis. Finally, due to, e.g., different modeling decisions and data/concepts used, comparing underlying models and obtained results would be challenging.

4 Vision and High-Level Goals

Our aim is to offer a hierarchy of DSMLs encompassing a *reference DSML* (rDSML) including concepts for conducting LCA, which are later refined to specific industrial domains with an increasing level of specificity, up to certain enterprises as *Enterprise-Specific Modeling Languages*, cf. Fig. 1. Thus, within the offered hierarchy, both companies would have an access to a domain-specific language for conducting LCAs in their industry, suited to their needs, supporting them in conducting the desired assessment, and ensuring required comparability and transparency of achieved results.

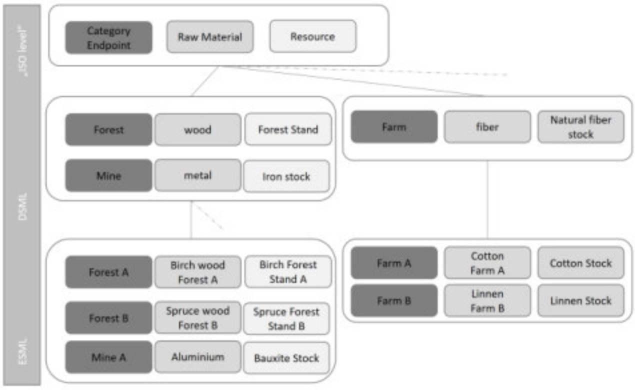

Figure 1. Vision: A hierarchy of DSMLs

And so, while the rDSML provides generic ISO concepts, the refined DSMLs provide specific, semantically rich concepts with a wide range of properties appropriate to describe the corresponding domain. These concepts store relevant 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 (domain-specific) knowledge on how to conduct the desired analysis. This knowledge encompasses, among others, a set of impacts, their indicators, assessment methods, requirements regarding the data to be used, as well as, if possible, the required data itself.

The hierarchy allows both companies to conduct the analysis on the local and global scale on different levels. This means that they can use 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 stakeholder groups, e.g., to identify impacts on a specific ecosystem caused by water extraction from a lake (environmental), or to identify social problems like excessive working hours at a specific site. This would enable stakeholders' engagement and provide the possibility to identify and address concrete problems. Finally, guidelines (e.g., for interpretation and presentation) are embedded into the language, enriched by a visual notation following cognitive principles of information design to facilitate communication and understanding of the process and its results.

In order to: (1) allow for application of concepts from different *impact assessment methods* in tandem, (2) allow users to access all classification (specificity) levels they are interested in, as well as (3) ensure the comparability of achieved results (e.g., by models explicating assumptions and information about the system boundaries, cf. [10]), all of the DSMLs in the hierarchy are integrated. This integration is achieved through the application of the same language architecture, through the refinement of concepts from the reference DSML (vertical integration), and also through the definition of aligning horizontal relationships between concepts. In consequence, the hierarchy of DSMLs offers the required transparency for evaluating and comparing the achieved assessment of product systems, cf. [6]. In addition, the presented hierarchy is adaptable, meaning that once new developments (e.g., new *impact assessment methods*) are known, they may be accounted for within the appropriate DSML.

Currently, there exists a wide range of LCA tools: generic expert tools and specialized ones focusing on specific areas [24]. Considering that the conflict of standardization versus extensibility is one of the major challenges discussed in LCA literature [5,6,14], this hierarchy of DSMLs should allow for a new generation of LCA tools enabling users to build semantically rich models while reusing existing concepts. In particular, the resulting tools should have flexible architecture allowing for integration of standards and best practices (on higher levels), while providing extensibility to account for different application scenarios and methodological advancements (on lower levels).

To summarize, the hierarchy of DSMLs is to support the following high-level goals: (G1) Provide a support for a wide range of different perspectives prospective users may be interested in, by providing a hierarchy of vertically and horizontally integrated DSMLs; (G2) Offer semantically rich concepts and required information to support the assessment process; (G3) Account for existing standards and branch-specific methods in a way to ensure comparability of achieved results, (G4) Support conducting analysis on a local and global level; (G5) Support extensibility and adaptation to account for the relevant changes; and (G6) Support both the productivity and reuse of the approach.

5 Requirements

In line with the method followed [16], based on the identified goals (Sect. 4), a set of identified application scenarios, analysis of LCA methods and instruments, as well as problems reported in the literature, e.g., [5,6,14], we have defined a set of requirements towards (1) the scope of the targeted solution (i.e., concepts and functionalities required to conduct LCA in different domains), and towards (2) the language architecture of the targeted artifact. Due to the space restriction, we discuss a few selected requirements towards language architecture only.

R1: Accounting for a hierarchy of professional terminology encompassing various classification levels. *Rationale*: Our goal is to provide support for a wide range of different perspectives of prospective users (cf. G1). Considering that different users consider involved concepts at different levels of specificity, there is a need to account for a hierarchy of professional terminology. For instance, the LCA concepts such as endpoint and raw material can be of interest on different levels of detail, e.g., as a raw material as such, as Plate of Wood, as Plate of Birch Wood only, or as instances of the latter, i.e., some specific forests. Therefore, the language should provide information on relevant aspects accounting for generic terms, their types, categories and instances. *Discussion:* In the current practice DSMLs are built by defining a meta model using some general-purpose modeling language, e.g., Meta Object Facility in UML [12]. This meta model describes concepts that users may use to create models. Thus, a modeling language usually encompasses two layers: specification (i.e., the definition of a DSML) and language application, cf. [25]. However, in our case, the language itself spans an arbitrary number of classification levels, not only two. Therefore, instead of emulating several meta-levels within two levels, or using artificial workarounds [26], a more natural way to define the desired hierarchy is the use of multi-level modeling [27]. Multi-level modeling refers to a language architecture that allows for an arbitrary number of classification levels being represented within a single body of model [27].

R2: Facilitating horizontal and vertical integration. *Rationale*: A rapid growth of various initiatives both horizontally (e.g., ecological and social aspects) and vertically (e.g., specific methods for some types of resources/some industries) causes a need for integration as well as ensuring comparability between achieved results (cf. G1 and G3). *Discussion:* In order to support integration (and also avoid redundancy), language users should be able to state what they know as soon as they know it, cf. [28]. For instance, already on the level of a rDSML encompassing, e.g., a concept *NaturalResource*, we would like to state that on the instance level (which is a few classification levels below) an attribute summing up production per year will be applicable. To be able to define it however, we would need a deep instantiation mechanism [27] that allows us to define some properties on a higher level and defer their instantiation to some not directly proceeding classification level. Such a mechanism is offered by multi-level modeling approaches and is not supported by traditional ones.

R3: Providing support for productivity of modeling and reuse. *Rationale*: The scenario indicates that a wide range of aspects of a concept (e.g., “resource”) should be accounted for in an integrated manner. Furthermore, their description should be rich enough to support various analyses regarding, e.g., their general character (e.g., hazardous) or specific characteristics (e.g., the rate of growth, societal relevance) [5, 19]. To this aim, we require domain-specific concepts with a rich set of attributes [29]. This is to support modeling productivity, i.e., productivity of creating, analyzing and modifying models (i.e., the time needed to accomplish those tasks [30]). However, on the other hand, at the same time we want to provide a set of generic concepts, so that they may be reused in a wide range of (not yet known) scenarios, cf. G6. *Discussion:* If we are modeling using the conventional approach and are limited to two classification levels, we need to face a well-known conflict between the level of reuse and modeling productivity [12]: the more specific a concept is, the lower range of reuse there is. On the other hand, the more general the concept is, the wider is the range of reuse, but also the lower is the productivity of modeling. As already discussed by other authors, e.g., [12], a satisfactory solution to this conflict comes with a multi-level modeling language architecture where we can account for both generic and specific concepts at the same time through multiple classification levels and, on demand, select those that suit our modeling needs.

R4: Incorporating relevant knowledge within the language. *Rationale*: Conducting a life-cycle assessment requires, among others, (1) information on potential impacts, their indicators as well as reference values; and (2) a level of expertise and experience that may not be available in many organizations [5, 19]. *Discussion:* Incorporating relevant knowledge within the language specification implies, among others, assigning states to (meta) classes (e.g., stating that the value of recycling code for Plate of Wood is 'FOR'). This however, is impossible in the traditional language architecture due to the iron rule of the type/instance dichotomy [12]. Therefore, we would need to always provide this information on the level of language application only. Considering it, application of multi-level modeling seems reasonable, as it offers "clabjects" [27], i.e., concepts having the characteristics of classes (defining a structure) and objects (having a state).

R5: Equipping models and their elements with behavior. *Rationale*: (Meta) classes (e.g., *NaturalResource*) have features that need to be derived or calculated based either on the content of the model (e.g., calculation of an impact profile), or based on the data acquired from external sources [7] (e.g., to obtain some environmental data for the needs of the assessment calculation), cf. G2. In addition, we would like to execute operations (automated analysis) on model elements not only on the level of objects, but also on the level of classes (e.g., calculation of global and local impact on the forest, cf. G4). *Discussion:* While it is possible to define language concepts having attributes constituting abstractions over desired data, the problems with the (automated) acquisition/calculation of those values emerge. The reason for this is not necessarily connected with the modeling paradigm as such, but rather with the programming languages used to create modeling tools. Modeling tools are usually developed using mainstream object-oriented programming languages, which feature only one classification level. Thus, types or even meta types are represented as objects by overloading the M_0 level of a programming language [28]. Therefore, a common representation of code and model is not possible and a model-code synchronization is required [28]. Thus, not only a recompilation step of modeling tools is required whenever we want to change something in the language specification (cf. R6), but also equipping model elements with operations (e.g., allowing to retrieve actual state of the environment) is hardly conceivable [28,31]. Therefore, a satisfactory solution seems to be the application of an integrated modeling and programming, i.e., using a modeling approach coming with a language execution engine.

R6: Ensuring extensibility and adaptability of the hierarchy without losing a corresponding tool support. *Rationale*: As already mentioned, there is a need to account for new developments and additional knowledge (G5), e.g., on impact caused by different substances. Therefore, the proposed hierarchy should be easily extensible and adaptable [19]. As an appropriate tool support is necessary to conduct the assessment process, those changes should not lead to losing this support. *Discussion:* If we decide to use a conventional approach and a tool based on the semantics of dominant object-oriented programming languages, even if users would have an access to language specification and would be able to extend it and adapt it to their needs, a recompilation step would be required to account for the changes in the corresponding tool [32]. A satisfactory solution to this problem again comes with the application of the multi-level approach, where the border between language specification and application is blurred, implying that by changing a multi-level model, a user may adapt the language to his/her needs [30]. If we use the integrated modeling and programming environment, also the tool-support will not be lost, cf. [28].

6 A Hierarchy of DSMLs for LCA

As discussed in the previous section, when it comes to the application of conventional two-level modeling paradigm, although its application is *technically* possible, it imposes limitations, which hinder us from delivering a satisfactory language specification, i.e., a solution without workarounds, overloaded levels, model redundancy and accidental complexity [12, 26, 33]. Therefore, we turn our attention to multi-level modeling with an integrated programming environment to propose a hierarchy of DSMLs.

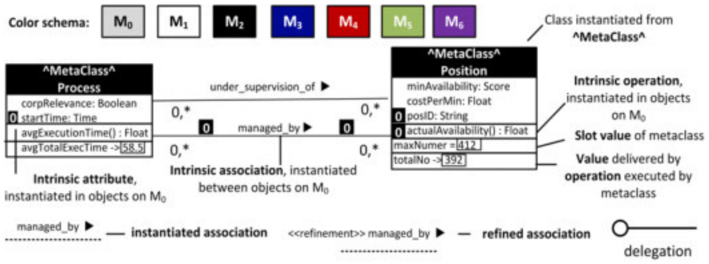

Figure 2. Concrete syntax of FMML^X, based on [12]

Multi-level modeling refers to a language architecture that allows for an arbitrary number of classification levels which are represented within a single body of model content, cf. [27]. As there is no strict division between language specification and language application, all languages (generic, regional, local ones) are integrated in one language architecture and thus, users can access all classification levels they are interested in. We can also benefit from such features as relaxed type/instance dichotomy, deferred instantiation, or defining level-crossing relations, cf. [31]. To illustrate the prospects of multi-level modeling, let us consider an excerpt of a multi-level model of resources created using a Flexible Meta Modeling and Execution Engine (FMML^x) [12]. We have selected FMML^x for our design due to the fact that, to the best of our knowledge, FMML^x as the only one has an integrated language execution engine offered by a supporting tool XModeler [12]. This feature allows us, among others, to equip models with behavior and provide support for computational analysis. Whereas the detailed description of FMML^x may be found in [12, 28], Fig. 2 presents its concrete syntax. Apart from the "traditional" modeling constructs such as classes, attributes, operations and relationships, it is possible to defer an instantiation of all modeling constructs by assigning them so called level of *intrinsicness*, which tells at which level of classification a given property will be instantiated.

Fig. 3 presents an excerpt of the designed hierarchy of DSMLs in the form of a multi-level model. Please note that for readability purposes, only selected concepts, selected attributes and operations derived from our scenario assigned to different levels of classification are presented. By supporting multiple classification levels, FMML^x offers the possibility to define and use concepts that correspond directly with the desired level of details (R1). Thus, we have a possibility to account for the fact that such concepts as *NaturalResource* or *RawMaterial* span multiple levels of classifications with categories, types and instances, cf. ① Fig. 3, without the need for overloading some level or applying some other workaround. At each level of classification, we have the possibility to express relevant information, making the model semantically rich (thus, we support the productivity of modeling as well as enable various analyses), and at the same time facilitate its reuse (R3). Regarding the latter, consider, e.g., attributes, operations and relationships defined for the concept *NaturalResource* (M_3), which reflect the domain knowledge derived out of the scenario on this classification level about this concept. Please note that a majority of those characteristics will be instantiated (i.e., assigned with values) only a few classification levels below (cf. the assigned level of intrinsicness). For example, the attribute *sum production per year local* (sumProdPyL) is instantiated in the local stocks of natural resources, be they stocks of cotton residing on farms or stocks of caolinite (mineral used for aluminum) residing on mines ②.

Now, while we move along the created hierarchy ③ (e.g., the chain starting from *RawMaterial* (M_4) via *PlateOfWood* (M_3) up to a *specific Plate of Wood* (M_2) and its instances), on the one hand, we instantiate the concepts, i.e., the relevant attributes are assigned with values (e.g., ④ the recycling code which is the same for all wooden products, and which is assigned on level M_3, cf. R4) and relevant operations aggregating, calculating or acquiring data from external sources may be executed (e.g., cf. ⑤, *calculateConsumptionWoW*(), or *sumAvailableResources*() defined for *NaturalResource*, M_4, cf. R5). On the other hand, we specialize those concepts, i.e., additional attributes, operations and relationships may be added to make concepts more and more specific (cf. ⑥, additional attributes defined for the concept *AbioticResource*, M_2). Furthermore, not only attributes are getting refined while moving along the hierarchy. This applies also to defined relationships. For instance, while we define that a *RawMaterial* (M_4) *stems from* some *NaturalResource* (M_3) and defer instantiation of this relationship to M_0, on the level of *CottonSheet* (M_2) we refine this relationship, and state that it can only stem from concepts instantiated from *CottonStock* (M_1) ⑦.

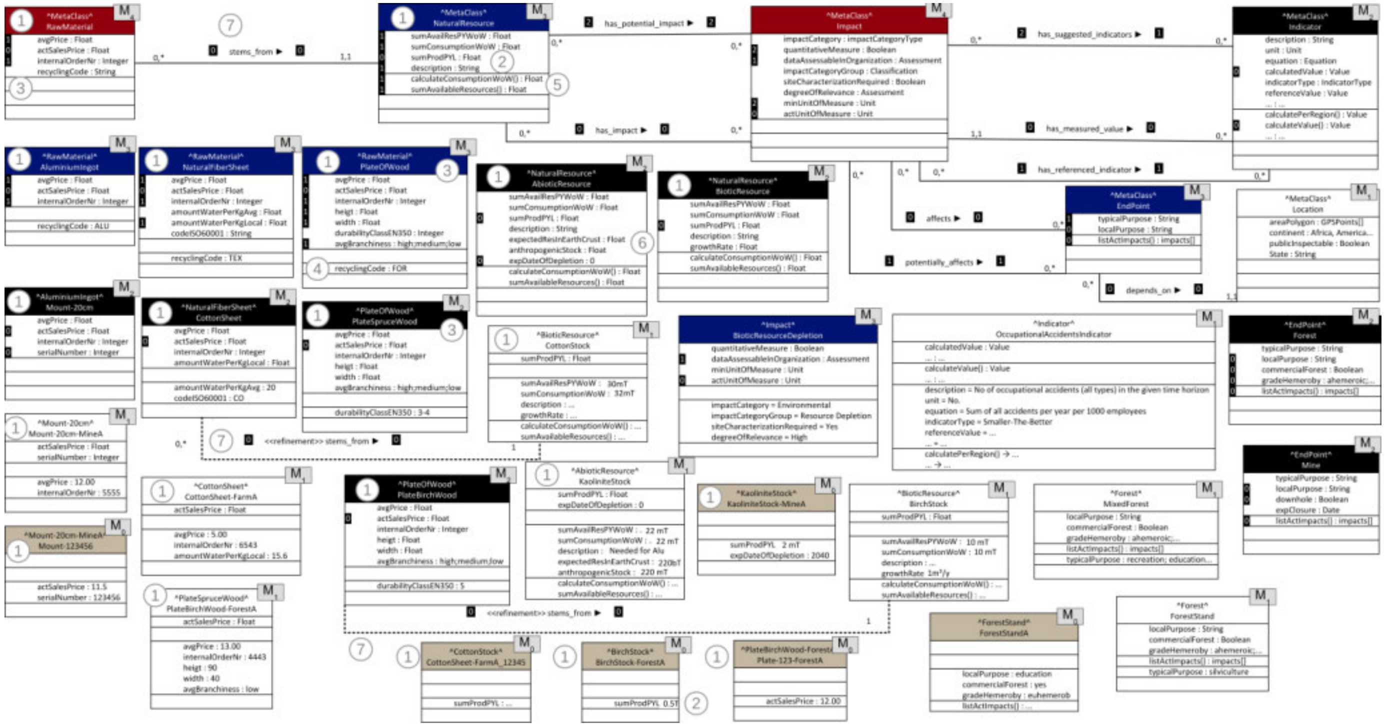

Figure 3. An excerpt of the designed hierarchy of DSMLs

The designed hierarchy fulfills the requirements (cf. Sect. 5) as summarized in Tab. 3. However, please note that the presented multi-level model may seem to be too complex for users [34]. The complex representation serves to exemplify how users can access concepts underlying their DSML and thus, benefit from greater transparency.

Table 3. Summary of evaluation against the requirements

Req.	*Evaluation and scenario-dependent solution*
R1	Concepts relevant for SD can be refined to domain specific concepts (e.g., Raw Material to Natural Fiber Sheet) up to enterprise specific concepts (types of Cotton Sheet) and instances.
R2	Domain-dependent attributes like recycling codes for Raw Material can be set once. Derived concepts on deeper levels are related to this information, but can be extended, as the vertical integration of Raw Material shows.
R3	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).
R4	Thanks to relaxed type/instance dichotomy we may assign state to classes, and thus, e.g., state what is the recycling code for a Plate of Wood.
R5	As in FMML^x a class is an object [12], operations can be not only specified for classes but also executed on them (e.g., calculateConsumptionWoW()).
R6	Thanks to a common representation of model and code provided by XModeler [12], a multi-level model may be extended without a need for recompilation.

7 Conclusion

In this paper, we argue that in order to support the LCA of products in a satisfactory manner, there is a need to design a hierarchy of DSMLs. To this aim we presented in this paper the clarification of the goals and scope of such a hierarchy, the requirements towards the language architecture as well as an excerpt of the current state of the resulting hierarchy of DSMLs. We also pointed that the results of our work can be used for building a new generation of LCA tools.

A few important limitations of our work need to be mentioned. Firstly, although the results of our work applied to different scenarios seem to be promising, they also point to the need for further extensions. Namely, additional work is required to incorporate further domain-specific LCA methods and techniques, and ensure integration to existing databases. In addition, as our ultimate goal is to support analysis targeting strong sustainability, or supporting at least a substitution strategy, cf. [10], there is a need to further extend the ecological, economic, and social aspects accounted for.

Secondly, additional work is required to reduce the complexity of a resulting multi-level model. While the complexity of multi-level models is not a novel phenomenon, cf. [31, 34], it should be addressed since it makes the models difficult to interpret, thus potentially inhibiting the added analysis capabilities we aim at. Here the integrated modeling and programming environment XModeler [12] comes into play with the promise of being able to define various perspectives adjusted to the information needs of prospective users. Finally, it has to be remarked, that further limitations need to be accounted for which are inherent to LCA itself. They comprise boundary issues as described by Hovorka et al. [35], uncertainty regarding ecological causes and effects [36], as well as the appropriateness of a systems' perspective for social aspects [37].

References

1. Melde, T.: Nachhaltige Entwicklung durch Semantik, Governance und Management. Springer, Wiesbaden (2012)
2. Finkbeiner, M., Inaba, A., Tan, R., Christiansen, K., Klüppel, H.J.: The new international standards for life cycle assessment: ISO 14040 and 14044. Int J of LCA 11(2), 80-85 (2006)
3. ISO: DIN EN ISO 14044 - Umweltmanagement - Ökobilanz - Anforderungen und Anleitungen (2006)
4. 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, pp. 314-322 (2018)
5. Hellweg, S., Milà i Canals, L.: Emerging approaches, challenges and opportunities in life cycle assessment. Science 344 (6188), 1109-1113 (2014)
6. Kuczenski, B., Marvuglia, A., Astudillo, M.F., Ingwersen, W.W., Satterfield, M.B., et al.: LCA capability roadmap-product system model description and revision. Int J of LCA 23(8), 1685-1692 (2018)
7. Kalverkamp, M., Karbe, N.: Comparability of life cycle assessments: Modelling and analyzing LCA using different databases. In: Pehlken, A., Kalverkamp, M., Wittstock, R., eds.: Cascade Use in Technologies 2018, pp. 51-63. Springer (2018)
8. Gardner, T.A., Benzie, M., Börner, J., Dawkins, E., Fick, S., Garrett, R., Godar, J., Grimard, A., Lake, S., Larsen, R.K., et al.: Transparency and sustainability in global commodity supply chains. World Development (2018)
9. 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)
10. Nolte, M., Kaczmarek-Heß, M.: Product life-cycle assessment in the realm of enterprise modeling. In: The Practice of Enterprise Modeling 2017, pp. 187-202 (2017)
11. Jacobs, M.: Sustainable Development as a Contested Concept. In: Dob, A., ed.: Fairness and Futurity: Essays on Environmental Sustainability and Social Justice, pp. 21-45. Oxford University Press, Oxford (1999)
12. Frank, U.: Multilevel modeling - toward a new paradigm of conceptual modeling and information systems design. BISE 6(6), pp. 319-337 (2014)
13. Becker, J., Janiesch, C., Seidel, S., Brelage, C.: A framework for situational and evolutionary language adaptation in information systems development. In: Advances in Information Systems Development, pp. 103-113. Springer (2007)
14. Unger, N., Beigl, P., Wassermann, G.: General requirements for LCA software tools. Institute of Waste Management, BOKU–University of Natural Resources and Applied Life Sciences, Vienna Austria (2004)
15. Österle, H., et al.: Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. Zeitschrift für betriebswirtschaftliche Forschung 62(6), 664-672 (2010)
16. Frank, U.: Outline of a Method for Designing Domain-Specific Modelling Languages. ICB Research Report 42, University of Duisburg-Essen, Essen (2010)
17. Finkbeiner, M.: The International Standards as the Constitution of Life Cycle Assessment: The ISO 14040 Series and its Offspring, pp. 85-106. Springer (2014)

18. Bare, J.C., Hofstetter, P., Pennington, D.W., De Haes, H.A.U.: Midpoints versus endpoints: the sacrifices and benefits. Int J Life Cycle Assess 5(6), 319 (2000)
19. Finkbeiner, M., Ackermann, R., Bach, V., Berger, M., Brankatschk, G., Chang, Y.J., et al.: Challenges in Life Cycle Assessment: An Overview of Current Gaps and Research Needs, pp. 207-258. Springer (2014)
20. Andrews, E.S., Barthel, L.P., Beck, T., Benoît, C., Ciroth, A., Cucuzzella, C., Gensch, C.O., et al.: Guidelines for Social Life Cycle Assessment of Products. United Nations Environment Programme (2009)
21. Onat, N.C., Kucukvar, M., Halog, A., Cloutier, S.: Systems thinking for life cycle sustainability assessment: a review of recent developments, applications, and future perspectives. Sustainability 9(5), 706 (2017)
22. DIN, E.: 350:2016-12. Durability of wood and wood-based products - Testing and classification of the durability to biological agents of wood and wood-based materials. Deutsches Institut for Normung (2016)
23. DIN: 6000-1:2001-05. Textile Faserstoffe - Teil 1: Naturfasern und Kurzzeichen. Deutsches Institut for Normung (2001)
24. Hitchcock, D., Schenk, R., Gordy, T.: Directory of Sustainability Life Cycle Assessment Tools. ISSP (2011)
25. de Lara, J., Guerra, E.: Domain-specific textual meta-modelling languages for model driven engineering. In: Vallecillo A., Tolvanen JP., Kindler E., Störrle H., Kolovos D. (eds) Modelling Foundations and Applications. ECMFA, pp. 259-274. Springer, Berlin (2012)
26. Atkinson, C., Kühne, T.: Reducing accidental complexity in domain models. SoSyM 7(3), 345-359 (2008)
27. 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)
28. 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, ceur-ws.org, pp. 3-24 (2016)
29. Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. SoSyM 13(3), 941-962 (2014)
30. Frank, U.: Power-modelling: Toward a more versatile approach to creating and using conceptual models. In: BMSD'14, pp. 9-19 (2014)
31. Kaczmarek-Heß, M., de Kinderen, S.: A Multilevel Model of IT Platforms for the Needs of Enterprise IT Landscape Analyses. BISE 5 (2017)
32. Atkinson, C., Gerbig, R., Fritzsche, M.: A multi-level approach to modeling language extension in the enterprise systems domain. IS 54, 289-307 (2015)
33. Lara, J.D., Guerra, E., Cuadrado, J.S.: When and how to use multilevel modelling. ACM Trans. Softw. Eng. Methodol. 24(2), 12:1-12:46 (December 2014)
34. Kaczmarek-Heß, M., Nolte, M., Fritsch, A., Betz, S.: Practical experiences with multi-level modeling using FMML^x. In: 5th intl. Works. on Multi-Level Modelling (2018)
35. Hovorka, D.S., Labajo, E., Auerbach, N.: Information systems in environmental sustainability: Of cannibals and forks. In vom Brocke, J., Seidel, S., Recker, J., eds.: Green Business Process Management, pp. 59-72. Springer, Berlin (2012)
36. Adensam, H., Ganglberger, E., Gupfinger, H., Wenisch, A.: Wieviel Umwelt braucht ein Produkt? Österreichisches Ökologie Institut, Wien (2000)
37. Schultze, U.: What kind of world do we want to help make with our theories? Information and Organization 27(1) 60 - 66 (2017)