Towards Hybrid Semantics of Enterprise Modeling Languages

Richard Braun and Werner Esswein
TU Dresden, Chair for Wirtschaftsinformatik, esp. System Development, 01062 Dresden, Germany

Keywords: Enterprise Modeling, Meta Modeling, Formal Semantics, Material Semantics, Behavioral Semantics.

Abstract: Enterprise Modeling Languages (EMLs) are generally perceived as conceptual modeling languages having a formal syntax and informal semantics. The non-formality of semantics is mainly caused by the materiality of the addressed domain (enterprises and its related aspects) and the resulting personal interpretation of syntactical constructs. However, EMLs may also explicitly define invariant interpretations in the sense of possible model executions or the definition of domain-specific restrictions. It is therefore promising to address a possible amalgamation of material semantics and formal semantics in order to provide an integrated and comprehensive semantic specification of EMLs. This position paper introduces and motivates the topic by systematizing and consolidating approaches from both fields and introduces a framework for so-called hybrid semantics on the meta model layer. Further, the general relevance of semantics and semantic specifications in EMLs is emphasized and prospective research challenges are proposed.

1 INTRODUCTION AND MOTIVATION

1.1 Semantics in Enterprise Modeling Languages

Enterprises are multifarious, heterogeneous socio-technical organizations, whose components are interrelated within a complex system on different abstraction levels (Vernadat, 2003). Enterprise Modeling (EM) aims to conceptualize, abstract and represent parts and aspects of enterprises by creating conceptual models in order to foster communication between involved stakeholders and enable an integration of static, procedural and functional dimensions (Lankhorst, 2009; Frank, 2013; Frank, 2014; Maes and Poels, 2007).

Enterprise Modeling Languages (EMLs) serve as languages for the construction of respectively needed enterprise models. The range of EMLs can be divided into integrated, domain-independent approaches like ArchiMate (Lankhorst et al., 2009), ARIS (Scheer and Nüttgens, 2000) or MEMO (Frank, 2014), purpose-specific languages like BPMN (processes (OMG, 2011)) or KAOS (risk management (Heaven and Finkelstein, 2004)) and dedicated, domain-specific approaches like CPmod (Burwitz et al., 2013).

Most EMLs constitute as semi-formal modeling languages (Wand and Weber, 2002), generally featured by a formal syntax and informal semantics (Karagiannis and Kühn, 2002). Informality of semantics primarily results from the fact that EMLs address the modeling of enterprises and related aspects, which are usually not entirely formalizable (Pfeiffer and Gehlert, 2005), as they mainly refer to real-world concepts (Lindland et al., 1994, p. 44). This can be also referred as ontological semantics (Harzallah et al., 2012, p. 489). EMLs cover a practically unlimited set of concepts (Anaya et al., 2010, p. 101). Semantics represents the meaning of a particular sign system (syntax) in regard of a semantic domain (Overhage et al., 2012, p. 22).

Consequently, stakeholders have different conceptualizations and understandings of particularly addressed real-world things (van der Linden, 2015). Also specific contexts (Bjekovic et al., 2014) or lexical ambiguities (Delfmann et al., 2009) may determine different interpretations of one and the same meta model construct. Besides this immanent feature of subject-dependent reference between enterprise models and semantic constructs (Rosemann et al., 2004; Guizzardi, 2007; Opdahl and Henderson-Sellers, 2002), invariant interpretation is explicitly desired in some enterprise models for different reasons (Bork and Fill, 2014, p. 3400). Besides, it could be explicitly intended to limit the variance of interpreting enterprise models in order to evolve and ensure a common, coherent understanding within a particu-

Braun, R. and Esswein, W.
Towards Hybrid Semantics of Enterprise Modeling Languages.
DOI: 10.5220/0005812504120420
In *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016)*, pages 412-420
ISBN: 978-989-758-168-7

lar domain (Frank, 2013, pp. 13-14) - for instance, in regulated and rather standardized domains like healthcare (Burwitz et al., 2013).

1.2 Research Objective and Research Context

It is therefore advisable to focus the current state of the art in regard of material semantics (Gehlert and Esswein, 2006) and formal semantics (Bork and Fill, 2014) in order to investigate reasonable integrations of these types. We therefore aim to provide a framework for differing types of semantics based on their characteristics. This is necessary, as the term *semantics* itself (Thalheim, 2012, p. 7) has several semantics within different research disciplines (Harel and Rumpe, 2004, pp. 67-69). For instance, semantics in formal modeling languages is understood as the invariant transformation of one valid model state to an other (Engels et al., 2000), while in semi-formal languages a mapping to semantic concepts is meant (Rosemann et al., 2004). This issue seems to be closely related to the immanent multi-disciplinarity in EM. The framework should further facilitate the identification of particular integration points and support the subsequent alignment and integration of the doubtless existing description techniques for semantics from different fields of research (van der Linden, 2015; Opdahl et al., 2012; Höfferer, 2007; Engels et al., 2000).

The motivation of the stated research goal comes from the rather poor semantic specification of modeling languages (Harel and Rumpe, 2004, pp. 67-69) and EMLs (Santos et al., 2013, pp. 690, 706), which hampers language comprehensibility (Guizzardi, 2007; Wyssusek, 2006) and provokes misunderstandings and semantic mismatches (van der Linden, 2015). Further, syntax and semantics (Natschläger, 2011b) or informal and formal aspects are often amalgamated (Gehlert et al., 2005). Especially in OMG specifications, syntactical statements and semantic explanations are often mixed and distributed over hundreds of specification document pages (Hausmann et al., 2005, p. 19), causing misunderstandings and contradictions (Natschläger, 2011b). This issue is even exacerbated due to the pure informality of semantic statements.

It is therefore necessary to elaborate and define a generic and integrated specification format for EML semantics in order to provide possibly precise semantics for meta model concepts, both in terms of variant as well as potentially invariant semantics. This in turn could facilitate semantic agreements in collaborative EM (Rospocher et al., 2008; Renger et al., 2008).

While the formal syntax of EMLs is fundamental, we agree to Bjekovic et al. (2014) stating that semantics and pragmatics are the actual drivers of EML usage. Investigating material and formal semantics in the EM domain is therefore also highly relevant in terms of EML dissemination and revision, as semantic comparisons might better justify the necessity for EML dialects and language adaptations (Braun, 2015; Bjekovic et al., 2013). Generally, general-purpose or purpose-specific languages like BPMN or ArchiMate provide generic semantics, which are often specified or refined for particular contexts (Hamann and Gogolla, 2013, p. 501).

The remainder of this position paper is therefore as follows. Section 2 introduces formal semantics in detail, as the role of formality in EM is only little investigated so far. Section 3 then presents material semantics. Section 4 outlines the aimed framework and proposes several topics for following research. The paper ends with a brief conclusion in Section 5.

2 FORMAL SEMANTICS

2.1 Relevance from the Perspective of Enterprise Modeling

Formal semantics cover the unambiguous and subject-independent usage of linguistic expressions, which causes interpretive invariance (Guizzardi et al., 2002). Formal semantics is relevant within enterprise modeling for basically two reasons: Automation and analysis (Braun, 2015). *Automation-oriented* formal semantics indirectly refers to the real-world domain in the sense of automatable process execution in order to avoid failures and improve efficiency in a proactive manner. Functional or process-oriented languages are conducted for that purpose (Bork and Fill, 2014). BPMN aims to support process execution by textually describing their automatic interpretation (OMG, 2011, p. 435).

While automation owns real-world references (e.g., references to particular resources or machines), in *analysis-oriented* formal semantics the transformation of models into other models or formalizations is primal. For instance, deontic logic is applied to BPMN (Natschläger, 2011b) and other authors introduced specific operations *on* models for BPMN model version management (Pietsch et al., 2014). However, formal semantics is rarely discussed from the perspective of EM and there is a lack of specifying required formal aspects in the context of EMLs (Bork and Fill, 2014). Below, generic approaches for both specifying formal semantic domains as well as the formal semantic mapping are hence introduced in order to prepare a potential application for invariant parts of enterprise modeling.

2.2 Formal Semantic Domain

The semantic domain of formal semantics does not correspond to the real world, but rather to a constructed artificial world, which can be seen as solution layer that aims to solve problems that are relevant for real world problems but require further interpretation (Gehlert, 2007; Pfeiffer and Gehlert, 2005). In detail, a formal semantic domain often acts as kind of a projection surface for solving consistency issues of modeling languages, while the explicit real world reference is omitted (Engels et al., 2001, pp. 187-188).

Consequently, formal semantic domains are usually described by formal expressions (e.g., algebra or graph-based languages), whose interpretations are generally perceived as invariant (Tarski, 1936; Hausmann, 2003; Schobbens et al., 2006). Formal semantics are mainly discussed in the context of the UML (Engels et al., 2000), implicating a naturally strong focus on design-oriented approaches within Software Engineering (Hausmann et al., 2005, p. 5). Although an explicit and integrated specification of semantics is deeply required (Maoz et al., 2010, p. 2), a lack of formal specifications can be observed, e.g. for UML (Soltenborn and Engels, 2009, p. 7).

Formal semantics can be divided into behavioral formal semantics and static formal semantics (Engels et al., 2000).

Behavioral formal semantics describe executable things in the sense of transformations between states in order to model token flows or state transitions (Hausmann et al., 2005, p. 25), aiming to enable automatic interpretation of each model instance (Hamann and Gogolla, 2013, p. 489). Consequently, abstract state machines, Petri nets, automatons or finitie actions traces are leveraged as semantic domains for behavioral semantics (Engels et al., 2000; Hausmann et al., 2005; Maoz et al., 2010; Kossak et al., 2014). There are further some works regarding execution semantics of BPMN focusing workflow derivation (Lam, 2012). By default, BPMN only provides rather imprecise textual statements on this topic (OMG, 2011, p. 435).

Static formal semantics aims to represent valid and consistent static states. Static semantics is rarely specified in modeling languages and refers to denotational approaches. It is hence more interesting to limit a particular interpretation scope by introducing additional constraints (e.g., regarding union subsets in UML (Hamann and Gogolla, 2013)) or specifying the technical domain that should be represented. For instance, system models (Cengarle et al., 2014) are used as formal domains of syntactical domains specified with the MontiCore language (Maoz et al., 2010).

2.3 Formal Semantic Mapping

The actual formality of formal semantics results from the invariant relation between syntactical constructs and the semantics domain, which enables an a priori definition of the semantics (Messer, 1999, p. 99). Formal semantics can hence be understood as replacement of expressions of one language by expressions of an other formal language causing syntactical mappings (Fill, 2015, p. 44). This actually leads to a continual shift of unambiguous interpretation (Gehlert, 2007, p. 37), constituting as rule-based chain of transformations (Holten, 2003, p. 50), which typically leads to a final formulation of mathematic or algebraic statements (Hausmann et al., 2005, p. 10).

Hence, formal semantic mapping represents the specification of two formal syntaxes; generally constituting as operational or even axiomatic semantic annotations (Kühn, 2004, pp. 34-35). Surprisingly, most mappings are only informally specified (Hausmann et al., 2005, pp. 25-27). Formal approaches cover process algebra, linear equations, task graphs or functions. In contrast to these textual approaches, some researchers propose model-based approaches for specifying relations between MOF-based meta models (Hausmann, 2003).

Finally, the Dynamic Meta Modeling (DMM) approach has to be taken into account. DMM is a generic approach for the integrated specification of formal semantics and mappings of behavior describing modeling languages, e.g., state charts or activity diagrams (Engels et al., 2000). DMM only requires a well-defined meta model and can be implemented in three major steps: First, the meta model of a language is slightly extended. Then the behavior is formally specified and a transformation into a typed graph has to be conducted in order to realize the behavioral semantics (Engels et al., 2000). Therefore, syntactical expressions are coupled with pre conditions (events), rules for transformation (action) and post conditions (Soltenborn and Engels, 2009). If the pre conditions for a syntactical construct are fulfilled then particular rules are applied until the post conditions are satisfied, which represent a new valid model state.

3 MATERIAL SEMANTICS

Material semantics generally covers enterprise-related domains (Harel and Rumpe, 2004; Lindland et al., 1994), which cannot be described with formal syntax per se (Pfeiffer and Gehlert, 2005; Gehlert, 2007). This is of primal relevance in EM, as the entire formalization of enterprise-related domains is practically not possible (cf. (Searle, 1984, p. 6), (Gehlert, 2007, p. 29) and (Pfeiffer and Gehlert, 2005, pp. 111-112)). The reason for that lies in the lack of distinction between particular terms; especially regarding synonyms, which cannot be differentiated confidently and cause a not Turing-computable problem (Pfeiffer and Gehlert, 2005). Besides, different EML users usually have different understandings of EML meta concepts (van der Linden, 2015).

We divide semantics on meta model layers into the *semantic domain* and *semantic mappings* (Karagiannis and Kühn, 2002)[1]. This understanding is shared by numerous researches in the field of ontological analysis (Wand and Weber, 1993; Rosemann et al., 2004).

3.1 Material Semantic Domain

The *semantic domain* covers all kinds of things the language should be able to express (Harel and Rumpe, 2004, p. 68) and is conceptualized by domain constructs representing those things (Lindland et al., 1994, p. 44). Consequently, syntactic constructs are already pre-conceptualized and implicitly limit possible semantic references (Bjekovic et al., 2013; Wyssusek, 2004). Basically, material semantics covers what should be represented, i.e. those referred domain constructs that should be expressed (Esswein and Weller, 2007, p. 2004). A further distinction into static and behavioral material semantics as applied in the field of formal semantics (Hausmann et al., 2005) seems to be superfluous, as the actual impact of a syntactical construct becomes only relevant in its individual application (cf. pragmatic semantics (Bjekovic et al., 2014)). The underlying conceptualization (Guizzardi, 2007) is only the carrier or intermediate layer for that. In contrast, in formal semantics the semantics itself represents the application, independent of any intermediate layer.

Domain constructs can represent physical things as well as artificially constructed things of a domain (Malt, 1990; Bloom, 1998). Thereby, it is extremely important to note that the perception, cognition and interpretation of those things are subject-dependent and determined by multiple factors (Bjekovic et al., 2014; Esswein and Lehrmann, 2013; van der Linden, 2015).

We refer to this as *structural ambiguity*, as it finally refers to the addressed domain constructs. Some EM researches investigate dimensions of basal meta concepts from the enterprise domain and demonstrate their immanent variance (van der Linden and Hoppenbrouwers, 2012; van der Linden and van Zee, 2014; van der Linden, 2015). However, most EMLs implicitly base on positivistic assumptions (Bjekovic et al., 2014, p. 438) indicating a precise and invariant reference to domain concepts, which is insufficient in this regard. Rather the positions of constructivism (models as constructions of reality-related conceptualizations) or at least critical rationalism (models as reconstructions) seem to be more appropriate (Gehlert et al., 2005; Esswein and Lehrmann, 2013) in order to correspond to the immanent socio-pragmatic purpose of EMLs (Bjekovic et al., 2014, p. 433).

3.2 Ambiguous Interpretation

Current representation approaches address the actual description of material domains with the help of denotational approaches (Kühn, 2004, p. 34); namely pre-defined generic ontologies (Wand and Weber, 1993) or EM-related ontologies (Gehlert et al., 2005; Höfferer, 2007; Anaya et al., 2010; Fill, 2015). All these approaches contain natural language statements implying the lack of invariant interpretation (Thalheim, 2012). We refer to this as *lexical ambiguity*. It covers same understandings with differing labeling (e.g., synonyms) as well as different understandings with same labeling (e.g., homonyms).

The stated ambiguities implicate subject-specific understandings of material semantics, which is also referred as personal semantics (van der Linden, 2015). We refer to this as *actual semantics* covering the actual understanding of an EML (perhaps influenced by situational particularities (Henderson-Sellers, 2005; Braun and Esswein, 2014). In contrast, *stipulated semantics* constitutes as the actual semantics of the language designers, which hence act as initial base for language understanding in the sense of an original reference point for comparison (Rosemann et al., 2004).

Reducing the mentioned difference between stipulated and actual semantics of different stakeholders causes the issue of semantic consensus-finding (Harzallah et al., 2012, p. 501) and ontological commitment (Guizzardi, 2007, p. 25) within EMLs. Frank

---

[1] We omit the consideration of semantics on the model layer, which is referred as inherent semantics (Bork and Fill, 2014, p. 3403), since this paper explicitly aims to investigate semantics within the *meta model* layer.

(2013) claims that the semantics of domain-specific EMLs need to be invariant within a particular domain over a certain time period (Frank, 2013, pp. 13-14). Obviously, such common understandings seem to be realizable within a small modeler community. It becomes more difficult when addressing general-purpose languages (e.g., UML) of purpose-specific languages (e.g., BPMN), which are underspecified and intentionally generic for a certain degree by definition (cf. the *Pool* meta class in BPMN, for instance). Consequently, respective syntactical constructs own several semantic mappings. It becomes obvious that respective consensus finding approaches - on top of current representation formats - need to be elaborated and investigated in further research.

3.3 Material Semantic Mapping

Semantic mapping is the mapping of syntactical constructs to domain constructs (Karagiannis and Kühn, 2002). This mapping usually constitute as interpretive step (Wand and Weber, 1993) due to the immanent role of subject-dependent interpretation and missing. In EM relevant languages, the mapping is mostly informally specified (Bork and Fill, 2014, p. 3407). The field of ontological analysis of EMLs focuses on those mappings in order to evaluate completeness of a language in regard of a given ontology. For instance, the BWW ontology provides a projection for the mapping of syntactical concepts of EMLs by adapting the ontology of Bunge (Wand and Weber, 1993).

4 TOWARDS HYBRID SEMANTICS

4.1 Pragmatics and Motivation

Pragmatics is understood as the actual application of an EML in a specific context in order to solve or support a particular task (Lindland et al., 1994; Thalheim, 2012). Pragmatics in the field of EM is rarely investigated so far (Bjekovic et al., 2014). We hence propose to distinguish pragmatics in accordance to the recipient of the model that represents the major point of utility creation, since enterprises constitute as socio-technical information systems. Usually, the recipient is a *human* actor, who is expected to perform something by using the model. This could be done for reasons of pure documentation, which focuses the status quo, or in regard of design and engineering, which rather focuses desired states (e.g. designing a particular application system). If the recipient is a *machine*, then this machine is expected to solve a particular task that was technically specified before (e.g. model-based analysis). Respective value is hence basically created either by humans (variant, not determinable) or machines (invariant, determinable). Potential hybrids between both types have to be decomposed until a clear differentiation is possible. The creation of value requires the understanding of models and respective meta models. In case of expected human actions, material semantics are appropriate, while formal semantics are relevant for machine actions. Formal semantics can be also relevant for human actors, if interpretation invariance can be created based on some kind of consensus finding.

There may be cases in EM, where both human actors and machines are recipient of a specific model. For instance, in the field of clinical process management, clinical pathway models represent the process-oriented, multi-perspective and integrated view on treating patients and respectively required service processes (Braun et al., 2014). While the treatment process is naturally not formal, some service processes can be formally defined in order to execute automatable tasks. For instance, a particular resource has to be automatically reordered by an ERP system if the execution of a specific activity is completed. In contrast, this activity is part of a treatment process that is manually executed and the stated resource can have different (material) semantics for human actors: A physician perceives some resource as instrument for treatment and a controlling expert refers to certain financial aspects. Despite the different semantic references of each actor (ERP system, physician, controlling expert), the thing *resource* is mutual and provides a point for integration.

Therefore, it seems to be promising to discuss the specification of respectively required semantics in order to enable an integrated definition of semantics (cf. Section 1.2) and support the identification of such semantic integration points.

4.2 Framework

It becomes obvious that single semantic types are useful for different purposes in the context of EM. These types of semantics are therefore summarized in the light of EM by focusing their semantics and ontological positions concerning the real-world in Table 1. They are further presented in detail below. Thereby, also potential adaptations of formal semantics to originally non-formal issues from EM are briefly considered.

Behavioral formal semantics can be seen as appropriate means for the specification of executable

Table 1: Types of Semantics in the Context of EMLs.

	Behavioral Formal Seman.	**Static Formal Semantics**	**Material Semantics**
Principle	*Transitions* between two valid states by mapping to a formal system and specifying the *transformation rules*.	Definition of *validation rules* for a model state by mapping to a formal system.	Mapping syntactical constructs to subject-dependent *domain constructs*.
Purpose for EMLs	*Execution* of parts of the EML, i.e. some of its meta classes become executable. For instance, invariant machine operations (Fill, 2015) or the semantics of executable BPMN classes (OMG, 2011, p. 435).	*Restrictions* on single meta classes or constructs of classes; similar to syntactic semantics (Bjekovic et al., 2014, p. 439). Alternatively useful for on top *model analysis*, i.e. mapping to a formal target model.	Shaping a particular discourse and its basic conceptions (Proper et al., 2005); pragmatic semantics in the sense of relating syntax to *individual* conceptualizations and the context of a particular user (Bjekovic et al., 2014).
Invariance	Needs to be *invariant* and mapped to truth-apt, executable syntax.	*Invariant* for syntactical mapping, *variant* for ontological mapping.	*Variant* to differing degrees (cf. Section 3.2).

meta classes of an EML by specifying legal transformations between respective model states. It is traditionally useful for the specification of any kind of transformative and behavioral aspect in parts. This might be the case for accompanying aspects of process models, like resource-related aspects (Braun and Esswein, 2014, p. 49). More precisely, the flow through a particular process model might cause human decisions and is hence not invariantly interpretable. But the affected resource transformations could be invariantly interpretable if particular resource allocations or transformation processes are triggered and executed (see above). Although this could be also done by an integration of a separate formal language, it is more comprehensive to keep it in one EML definition and one EML semantics definition. Respective classes are therefore not solely descriptive, but contain executable semantics.

Behavioral formal semantics require invariant semantic mappings to a formal target model (syntax of the explaining modeling language, cf. Section 2.3). This indicates that a respective syntactical construct from the meta model should become subject of an ontological discussion in order to find respective consensus. After reaching an invariant semantic mapping to a particular domain construct, the domain construct should be mapped invariantly to a syntactical construct in order to specify respective transformations. Briefly speaking, the syntactical construct is mapped to the syntactical construct of an other language (Fill, 2015) after an explicit agreement on the respective reference domain is reached (Frank, 2013, pp. 13-14). Finally, a particularly instantiated model represents not only real-world things descriptively, but rather can *manipulate* parts of it (e.g., by coupling a particular machine control unit with respective model elements).

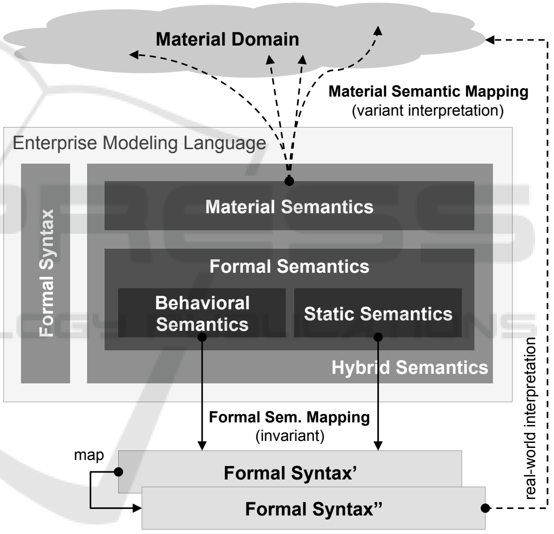

Figure 1: Outlining Hybrid Semantics in the Context of EML Specifications.

Static formal semantics might be applicable for the transformation of parts of the meta model into other formal models aiming to define particular analysis on top of a model (*external view*) on the one side (Natschläger, 2011a; Pietsch et al., 2014). On the other side, it can be used for specifying valid model states (*internal view*), which represent inherent rules and restrictions of a particular domain.

In contrast to behavioral formal semantics, transformation rules are not defined. Due to the required invariance between the syntactical constructs, respective consensus on the those relations is necessary, i.e. consensus on the syntactical meaning of meta model elements. This implies that a particular meta model element should only be invariant in regard of this relation, but can have several ontological mappings besides. Static formal semantics is hence view-dependent.

Material semantics are naturally useful for the explication of individual conceptualizations about particular discourse objects (Guizzardi, 2007; Bjekovic et al., 2014), indicating a particular agreement on related domain constructs (Lindland et al., 1994; Esswein and Lehrmann, 2013; van der Linden, 2015), which aims to enable a common and efficient communication about a domain. Material semantics are therefore primarily useful for supporting *externalization, documentation and communication* between individuals.

Figure 1 outlines the concept of hybrid semantics by splitting the EML semantics definition into parts for material and formal semantics, which are integrated by corresponding syntactical elements of the meta model. The framework also depicts the final relation of the type of formal semantics to the real-world.

5 CONCLUSIONS AND FURTHER RESEARCH

This position paper outlines the prospective definition of hybrid semantics for EMLs in order to explicitly integrate formal and material semantics in a comprehensive manner. The paper should be understood as starting point and contributes to the research community by structuring and framing different semantic approaches and understandings, which are relevant for EM. This could be seen as a first step for multi-disciplinary integration, tearing down still existing academic stonewalls between Information Systems Research (especially material semantics) and Software Engineering (especially formal semantics) in order to elaborate respective synergies for EML semantics. This explicitly covers the aspect of *using and applying* enterprise models and addresses finally pragmatics, which is still a poorly investigated topic that is largely omitted (Bjekovic et al., 2014).

Naturally, several topics need to be investigated in further research. We explicitly propose the following three topics.

Firstly, it is extremely important to specify the considered integration of material and formal semantics on the meta model layer. We therefore conduct two case studies in the field of manufacturing and healthcare in order to derive more insights and technical consequences on that issue.

Secondly, it is also important to determine a specification format for hybrid semantics. It is therefore planned to conduct the DMM approach (Engels et al., 2000) for formal aspects and ontology-based approaches for material aspects. A structural enhancement of these ontologies should be examined (van der Linden, 2015).

Thirdly, it is necessary to address the actual process of consensus finding as stated in Section 3.2, since a certain level of invariance is immanent for any kind of execution and automation. This includes the so far less discussed issue of explicitly specifying semantics of generic EMLs for specific contexts, domains or situations. This could lead to language dialects keeping the syntax stable and changing the semantic mappings, for instance (Braun and Esswein, 2014, pp. 47, 53).

ACKNOWLEDGEMENT

This research paper presents partial research results from the research project *SFB Transregio 96*, which is funded by the German Research Foundation (DFG).

REFERENCES

Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevičius, R., Opdahl, A. L., Panetto, H., and Verdecho, M. J. (2010). The unified enterprise modelling language—overview and further work. *Computers in Industry*, 61(2):99–111.

Bjekovic, M., Proper, H. A., and Sottet, J.-S. (2013). Enterprise modelling languages-just enough standardisation? In *BMSD*, pages 1–23.

Bjekovic, M., Proper, H. A., and Sottet, J.-S. (2014). Embracing pragmatics. *Conceptual Modeling*, 8824:431–444.

Bloom, P. (1998). Theories of artifact categorization. *Cognition*, 66(1):87–93.

Bork, D. and Fill, H.-G. (2014). Formal aspects of enterprise modeling methods: A comparison framework. In *Proceedings of the 47th Hawaii International Conference on System Sciences*, pages 3400–3409. IEEE.

Braun, R. (2015). Towards the state of the art of extending enterprise modeling languages. In *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development*, pages 394–402.

Braun, R. and Esswein, W. (2014). Classification of domain-specific bpmn extensions. In *The Practice of Enterprise Modeling*, volume 147 of *Lecture Notes of Business Information Processing*, pages 42–57.

Braun, R., Schlieter, H., Burwitz, M., and Esswein, W. (2014). Bpmn4cp: Design and implementation of a bpmn extension for clinical pathways. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 9–16. IEEE.

Burwitz, M., Schlieter, H., and Esswein, W. (2013). Modeling clinical pathways-design and application of a domain-specific modeling language. *Wirtschaftsinformatik Proceedings*.

Cengarle, M. V., Grönninger, H., and Rumpe, B. (2014). System model semantics of class diagrams. *arXiv preprint arXiv:1409.6635*.

Delfmann, P., Herwig, S., and Lis, L. (2009). Unified enterprise knowledge representation with conceptual models-capturing corporate language in naming conventions. *ICIS 2009 Proceedings*, page 45.

Engels, G., Hausmann, J. H., Heckel, R., and Sauer, S. (2000). Dynamic meta modeling: A graphical approach to the operational semantics of behavioral diagrams in uml. In *UML 2000 – The Unified Modeling Language*, pages 323–337. Springer Berlin Heidelberg.

Engels, G., Küster, J. M., Heckel, R., and Groenewegen, L. (2001). A methodology for specifying and analyzing consistency of object-oriented behavioral models. In *ACM SIGSOFT Software Engineering Notes*, volume 26, pages 186–195. ACM.

Esswein, W. and Lehrmann, S. (2013). About the need for semantically enriched reference models. In *Proceedings of the 19th Americas Conference on Information Systems*.

Esswein, W. and Weller, J. (2007). Method modifications in a configuration management environment. In *Proceedings of the Fifteenth European Conference on Information Systems*, pages 2002–2013.

Fill, H.-G. (2015). Metamodeling as an interface between syntax and semantics. In Glück, B., Lachmayer, F., Schefbeck, G., and Schweighofer, E., editors, *Elektronische Schnittstellen in der Staatsorganisation*, pages 43–50.

Frank, U. (2013). Domain-specific modeling languages: requirements analysis and design guidelines. In *Domain Engineering*, pages 133–157. Springer.

Frank, U. (2014). Enterprise modelling: The next steps. *Enterprise Modelling and Information Systems Architectures*, 9(1):22–37.

Gehlert, A. (2007). *Migration fachkonzeptueller Modelle*. Logos-Verlag.

Gehlert, A., Buckmann, U., and Esswein, W. (2005). Ontology based method engineering. *AMCIS 2005 Proceedings*, page 436.

Gehlert, A. and Esswein, W. (2006). Toward more rigor in ontological analyses. In *Proceedings of the 14th European Conference on Information Systems*, pages 984–994.

Guizzardi, G. (2007). On ontology, ontologies, conceptualizations, modeling languages, and (meta) models. *Frontiers in artificial intelligence and applications*, 155:18.

Guizzardi, G., Ferreira Pires, L., and Van Sinderen, M. J. (2002). On the role of domain ontologies in the design of domain-specific visual modeling langages.

Hamann, L. and Gogolla, M. (2013). Endogenous metamodeling semantics for structural UML 2 concepts. In Moreira, A., Schätz, B., Gray, J., Vallecillo, A., and Clarke, P. J., editors, *Model-Driven Engineering Languages and Systems - 16th International Conference, MODELS 2013, Miami, FL, USA, September 29 - October 4, 2013. Proceedings*, volume 8107 of *Lecture Notes in Computer Science*, pages 488–504. Springer.

Harel, D. and Rumpe, B. (2004). Meaningful modeling: what's the semantics of semantics? *Computer*, 37(10):64–72.

Harzallah, M., Berio, G., and Opdahl, A. L. (2012). New perspectives in ontological analysis: Guidelines and rules for incorporating modelling languages into ueml. *Information Systems*, 37(5):484–507.

Hausmann, J. H. (2003). Metamodeling relations-relating metamodels. *Metamodelling for MDA*, pages 147–161.

Hausmann, J. H., Heckel, R., and Sauer, S. (2005). *Dynamic Meta Modeling*. PhD thesis, University of Paderborn.

Heaven, W. and Finkelstein, A. (2004). Uml profile to support requirements engineering with kaos. *IEE Proceedings-Software*, 151(1):10–27.

Henderson-Sellers, B. (2005). Uml–the good, the bad or the ugly? perspectives from a panel of experts. *Software and Systems Modeling*, 4(1):4–13.

Höfferer, P. (2007). Achieving business process model interoperability using metamodels and ontologies. In *Proceedings of the 15th European Conference on Information Systems*, pages 1620–1631.

Holten, R. (2003). Integration von informationssystemen. *Wirtschaftsinformatik*, 45(1):41–52.

Karagiannis, D. and Kühn, H. (2002). Metamodelling platforms. In *EC-Web*, volume 2455, page 182.

Kossak, F., Illibauer, C., Geist, V., Kubovy, J., Natschläger, C., Ziebermayr, T., Kopetzky, T., Freudenthaler, B., and Schewe, K.-D. (2014). *A Rigorous Semantics for BPMN 2.0 Process Diagrams*. Springer.

Kühn, H. (2004). *Methodenintegration im Business Engineering*. PhD thesis, Universität Wien.

Lam, V. S. (2012). A precise execution semantics for bpmn. *IAENG International Journal of Computer Science*, 39(1):20–33.

Lankhorst, M. (2009). {Enterprise Architecture at Work: Modelling, Communication and Analysis (The Enterprise Engineering Series)}.

Lankhorst, M. M., Proper, H. A., and Jonkers, H. (2009). The architecture of the archimate language. In *Enterprise, Business-Process and Information Systems Modeling*, pages 367–380. Springer.

Lindland, O. I., Sindre, G., and Solvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software*, 11(2):42–49.

Maes, A. and Poels, G. (2007). Evaluating quality of conceptual modelling scripts based on user perceptions. *Data & Knowledge Engineering*, 63(3):701–724.

Malt, B. C. (1990). Features and beliefs in the mental representation of categories. *Journal of Memory and Language*, 29(3):289–315.

Maoz, S., Ringert, J. O., and Rumpe, B. (2010). A manifesto for semantic model differencing. In *Models in Software Engineering - Workshops and Symposia at MODELS 2010, Reports and Revised Selected Papers*, pages 194–203.

Messer, B. (1999). Zur interpretation formaler geschäftsprozeß-und workflow-modelle. In *Wirtschaftsinformatik und Wissenschaftstheorie*, pages 95–123. Springer.

Natschläger, C. (2011a). Deontic bpmn. In *Database and Expert Systems Applications*, pages 264–278. Springer.

Natschläger, C. (2011b). Towards a bpmn 2.0 ontology. In *Business Process Model and Notation*, pages 1–15. Springer.

OMG (2011). *Business Process Model and Notation (BPMN) - Version 2.0*. Object Management Group (OMG).

Opdahl, A. L., Berio, G., Harzallah, M., and Matulevicius, R. (2012). An ontology for enterprise and information systems modelling. *Applied Ontology*, 7(1):49–92.

Opdahl, A. L. and Henderson-Sellers, B. (2002). Ontological evaluation of the uml using the bunge–wand–weber model. *Software and Systems modeling*, 1(1):43–67.

Overhage, P. D. S., Birkmeier, D. Q., and Schlauderer, S. (2012). Quality marks, metrics, and measurement procedures for business process models. *Business & Information Systems Engineering*, 4(5):229–246.

Pfeiffer, D. and Gehlert, A. (2005). A framework for comparing conceptual models. In *Proceedings of the Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2005)*, pages 108–122.

Pietsch, P., Müller, K., and Rumpe, B. (2014). Model matching challenge: Benchmarks for ecore and bpmn diagrams. *arXiv preprint arXiv:1408.5693*.

Proper, H. A., Verrijn-Stuart, A. A., and Hoppenbrouwers, S. (2005). On utility-based selection of architecture-modelling concepts. In *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43*, pages 25–34. Australian Computer Society, Inc.

Renger, M., Kolfschoten, G. L., and de Vreede, G.-J. (2008). Challenges in collaborative modeling: A literature review. In *Advances in Enterprise Engineering I*, pages 61–77. Springer.

Rosemann, M., Green, P., and Indulska, M. (2004). A reference methodology for conducting ontological analyses. In *Conceptual Modeling*, pages 110–121. Springer.

Rospocher, M., Ghidini, C., Serafini, L., Kump, B., Pammer, V., Lindstaedt, S. N., Faatz, A., and Ley, T. (2008). Collaborative enterprise integrated modelling. In *SWAP*, volume 426. Citeseer.

Santos, P. S., Almeida, J. P. A., and Guizzardi, G. (2013). An ontology-based analysis and semantics for organizational structure modeling in the aris method. *Information Systems*, 38(5):690–708.

Scheer, A.-W. and Nüttgens, M. (2000). *ARIS architecture and reference models for business process management*. Springer.

Schobbens, P.-Y., Heymans, P., and Trigaux, J.-C. (2006). Feature diagrams: A survey and a formal semantics. In *Requirements Engineering, 14th IEEE international conference*, pages 139–148. IEEE.

Searle, J. R. (1984). *Minds, brains and science*. Harvard University Press.

Soltenborn, C. and Engels, G. (2009). Towards test-driven semantics specification. In *Model Driven Engineering Languages and Systems, 12th International Conference, MODELS 2009, Proceedings*, pages 378–392.

Tarski, A. (1936). Der wahrheitsbegriff in den formalisierten sprachen.

Thalheim, B. (2012). Syntax, semantics and pragmatics of conceptual modelling. In *Natural Language Processing and Information Systems*, pages 1–10. Springer.

van der Linden, D. (2015). Personal semantics of meta-concepts in conceptual modeling languages.

van der Linden, D. and Hoppenbrouwers, S. (2012). Challenges of identifying communities with shared semantics in enterprise modeling. In *The Practice of Enterprise Modeling*, pages 160–171. Springer.

van der Linden, D. and van Zee, M. (2014). On the semantic feature structure of modeling concepts: an empirical study. In *Business Informatics (CBI), 2014 IEEE 16th Conference on*, volume 2, pages 158–165. IEEE.

Vernadat, F. B. (2003). *Enterprise modelling and integration*. Springer.

Wand, Y. and Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. *Information Systems Journal*, 3(4):217–237.

Wand, Y. and Weber, R. (2002). Research commentary: information systems and conceptual modeling—a research agenda. *Information Systems Research*, 13(4):363–376.

Wyssusek, B. (2004). Ontology and ontologies in information systems analysis and design: A critique. *AMCIS 2004 Proceedings*, page 535.

Wyssusek, B. (2006). On ontological foundations of conceptual modelling. *Scandinavian Journal of Information Systems*, 18(1):article 9.