

Challenges of Modelling Landscapes

Pragmatics Swept Under the Carpet?

Marija Bjeković^{abc} and Henderik A. Proper^{abc}

^aPublic Research Centre Henri Tudor, Luxembourg, Luxembourg

^bRadboud University Nijmegen, Nijmegen, Netherlands

^cEE-Team¹, Luxembourg, Luxembourg

{marija.bjekovic, erik.proper}@tudor.lu

Keywords: model, modelling language, model integration

Abstract: In enterprise modelling, a wide range of models and languages is used to support different purposes. If left uncontrolled, this can easily result in a fragmented perspective on the enterprise, its processes and IT support. On its turn, this negatively affects traceability, the ability to do cross-cutting analysis, and the overall coherence of models. Different strategies are suggested to achieve model integration. They mainly address syntactic-semantics aspects of models/languages, and only to a limited extent their pragmatics. In actual use, the ‘standardising’ and ‘integrating’ effects of traditional approaches (e.g. UML, ArchiMate) erodes. This is typically manifested by the emergence of local ‘dialects’, ‘light weight versions’, as well as extensions of the standard to cover ‘missing aspects’. This paper aims to create more awareness of the factors that are at play when creating integrated modelling landscapes. Relying on our ongoing research, we develop a fundamental understanding of the driving forces and challenges related to modelling and linguistic variety within modelling landscapes. In particular, the paper discusses the effect of *a priori* fixed languages in modelling and model integration efforts, and argues that they bring about the risk of neglecting the pragmatic richness needed across practical modelling situations.

1 INTRODUCTION

Enterprise models play an important role in the design and operations of enterprises (Bubenko et al., 2010). More specifically, enterprise models can be used to study the current state of an enterprise, analyse problems with regard to the current situation, sketch potential future scenarios, design future states of the enterprise, communicate with stakeholders, manage change, etc. (Davies et al., 2006), (Bubenko et al., 2010) (Anaby-Tavor et al., 2010).

Next to the fact that enterprise models are created for different purposes, it is necessary to do so from different perspectives, such as business processes, value exchanges, products and services, information systems, etc. In the field of information systems engineering, the use of a multi-perspective approach has long since been advocated, e.g. (Wood–Harper et al., 1985) (Zachman, 1987). For enterprise

modelling, there is even a broader set of perspectives to consider (Frank, 2002) (Winter and Fischer, 2007) (Greefhorst et al., 2006) (Wagter et al., 2012).

The collection of models that jointly represent the different perspectives of one enterprise, are often expressed using different modelling languages, including UML (OMG, 2003), BPMN (OMG, 2008), ArchiMate (Jacob et al., 2012), i* (Yu and Mylopoulos, 1996), e3Value (Gordijn and Akkermans, 2003), SBVR (OMG, 2006), etc. Throughout this paper, we will use the term *enterprise modelling landscape*, or simply *modelling landscape*, to refer to the variety of models and corresponding modelling languages used in a specific enterprise modelling effort¹.

Since the models included in an enterprise modelling landscape provide different views on the *same* enterprise, it is quite natural to expect that the sets of models form a coherent whole; i.e. linked where relevant and consistent as a whole. A plethora of models and modelling languages may easily result in a frag-

¹The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University Nijmegen and HAN University of Applied Sciences (www.ee-team.eu).

¹The scope of a particular enterprise modelling effort can be only one project, cross-project considerations, entire enterprise etc.

mented perspectives on the enterprise, which is likely to have a negative impact on the traceability across different models, the ability to do cross-cutting analysis, to manage inconsistency² and to ensure overall coherence of e.g. the design of the enterprise. The fact that different models are usually expressed in different modelling languages makes it even harder to maintain the coherence.

The traditional approach of dealing with fragmentation in modelling landscapes is to create an integrated/unified modelling language, such as UML and ArchiMate. However, in actual use, one can observe how the ‘standardising’ and ‘integrating’ effect of such languages erodes. This typically manifests itself in terms of local ‘dialects’, ‘light weight versions’, or several extensions of an existing standard that are intended to deal with ‘missing aspects’. This point is illustrated by the advent of domain or purpose-specific (modelling) languages that allow for the creation of models that are tuned to the needs of specific domains or purposes. At the same time, there exists a number of approaches that aim to alleviate this fragmenting effect by assuring the links between the different languages definitions, see e.g. (Frank, 2002) (Vernadat, 2002) (Anaya et al., 2010). Typically, these links are defined based on the standardised definition of the language, in particular its semantics (as in e.g. (Anaya et al., 2010)).

As we will discuss throughout this paper, the drivers of language standardisation are predominantly of technical nature (Hoppenbrouwers, 2003). There are clear benefits of language standardisation. For instance, it is generally considered as a necessary condition for CASE tool development. In addition, it is a first step towards *automating* some model manipulations, e.g. model transformations (including code generation). Nonetheless, the potential benefits of standardised languages tend to be quickly generalised to the entire modelling endeavour. What goes practically unquestioned, in aiming for language standardisation, is whether fixed languages can be used at all in different modelling contexts and with different stakeholder groups. For instance, in (Kaidalova et al., 2012) (Bubenko et al., 2010) it is observed that the choice of formalism should be related the given modelling task and audience. For example, when the language chosen is rather too formal for stakeholders, it can hinder the modelling process.

We will argue, that standardising/fixing a modelling language leads to a situation in which the *pragmatic* richness that is needed across various modelling situations in practice is neglected. This

²Allow inconsistencies between models, e.g. due to different views by differing stakeholders, in a controlled way.

brings about the risk of *sweeping pragmatics under the carpet*. Indeed, various surveys (Malavolta et al., 2012) (Kaidalova et al., 2012) and empirical studies (Anaby-Tavor et al., 2010) (Karlsen, 2011) (Briand et al., 1995) (Elahi et al., 2008) reporting on practical experiences with enterprise modelling, point at the need for flexibility in modelling. At the same time they observe a lack in flexibility of tools and the underlying (fixed) languages to aptly fit the needs of specific modelling situations. This often leads to different levels of discipline in which the standard language is obeyed to, e.g. resulting in dialect-like variations of the original language (Bubenko et al., 2010) (Malavolta et al., 2012) (Briand et al., 1995) (Elahi et al., 2008) (Karlsen, 2011), or workarounds (e.g. using ad hoc notes and annotations) to compensate for the missing elements in the language/tool (Delen et al., 2005). This may even go as far as the use of home-grown, organisation-specific semi-structured models/languages instead of the standard notation (Anaby-Tavor et al., 2010) (Malavolta et al., 2012) (Karlsen, 2011).

In our view, this indicates a lack of fundamental understanding of the role of language in modelling, and more specifically, the place of fixed language in attempts to integrate models. In this paper, based on our ongoing research, we intend to shed more light on this topic. We will therefore start in Section 2 with a discussion of the potential benefits of standardising modelling languages. In Sections 3 and 4 we then explore the effects of their use in modelling and integration respectively, also identifying more explicitly the risk of *sweeping pragmatics under the carpet* of the modelling landscape. We then continue in Section 5 with a fundamental discussion of *models* and *modelling*. This understanding is used to then, in Section 6 discuss the role of modelling languages, and their potential standardisation. In the conclusion, we synthesise the insights and suggest the direction to explore more realistic strategies for creating and managing modelling landscapes.

2 STANDARDISATION

In our field, we typically deal with *linguistic models* (Karagiannis and Höfferer, 2006), i.e. models expressed in a modelling language. A further distinction can be made between textual and graphical languages (Harel and Rumpe, 2004). Given their significant usage in enterprise modelling efforts, our discussion focusses on graphical languages. Traditionally, a modelling language is defined in terms of an *abstract*

syntax, a concrete syntax and semantics.

The *abstract syntax* defines the basic elements and rules for creating models. The abstract syntax of graphical modelling language is commonly given by means of the meta-model. The meta-model actually represents the *conceptual foundation* of the modelling language, i.e. a specific classification of concepts to be used in discourse about the ‘world’ (Falkenberg et al., 1998). As such, the meta-model provides a particular *ontological position* filtering the view on the ‘world’ that one chooses to take (Falkenberg et al., 1998). It is also argued in (Falkenberg et al., 1998) that all other aspects of the modelling language depend on the concepts contained in the meta-model.

The *concrete syntax* or *notation* deals with the (visual) representation of the modelling language on the medium, by defining the visual symbols and rules for their combination (and their correspondence to the abstract syntax of the language). The medium itself can for example be restricted to a specific form, such as graphical, textual, or video, but the notation in general can also be restricted in terms of fonts, icons and layout rules. See e.g. (Moody, 2009).

The *semantics* deals with the meaning of a modelling language. The conventional way of defining semantics is in terms of a *semantic domain* and a *semantic mapping* (Harel and Rumpe, 2004). According to (Harel and Rumpe, 2004), the semantic domain captures the “*decisions about the kinds of things language should express*” (Harel and Rumpe, 2004, p. 68). The semantic mapping, in turn, establishes the correspondence from syntactic elements to the semantic domain. However, this approach to defining semantics is required for the mechanical manipulation of models, e.g. by computer tools, since they can only manipulate semantics in terms of syntactic representations (Harel and Rumpe, 2004). We propose to label this aspect of semantics as *syntactic semantics*. In the realm of human use of language, *meaning* is approached differently by taking into account the entire context in which the linguistic communication is embedded and the *function* of linguistic utterances in that context³. We propose to label this aspect of semantics as *pragmatic semantics*. When stakeholders use modelling language in modelling, they address the semantics from this perspective. We will discuss this topic further in Section 6.

As illustrated in Figure 1, the fixed i.e. *standardised* definition of the modelling language *a priori* identifies and restricts the *intended* sets of models the language allows to express. This also limits the sets

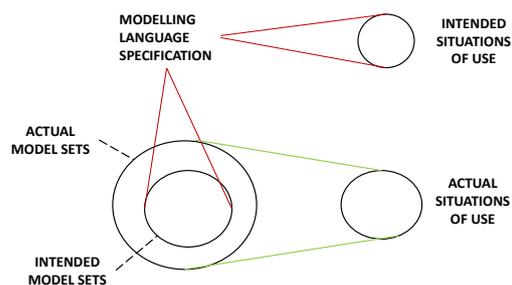


Figure 1: Intended and actual use of the modelling language

of *intended* modelling situations in which, by using a particular fixed language, a satisfactory model of the domain can be produced. Therefore, when *freezing languages* (Hoppenbrouwers, 2003), the designers of the language, implicitly or explicitly, restrict the intended use of a language. The more models a language can ‘produce’, the more *expressive* the language is. However, the actual *suitability* of the language is dependent on the particular modelling situation in which language is *used*. It refers to whether the modelling language allows to create the model of the domain such that it *satisfies* the needs of actual modelling situation (e.g. the level of detail in domain, coverage of specific aspects, specific form etc.). This is the area of *modelling pragmatics*. According to (Thalheim, 2012), it studies the use of languages in a particular modelling situation depending on the purposes and goals of models within a community of practice.

In most cases, graphical modelling languages are defined semi-formally, i.e. with explicit (and more or less formal) definitions of the (abstract and concrete) syntax. The conceptual foundation of the language may be defined at different levels of genericity (i.e. involving more or less generic concepts). This influences the intended model sets supported by the language. Also, different syntactic restrictions may be included in the language definition, further restricting intended model sets. The semantics is, however, usually given in an informal manner in the language specification, i.e. using natural language. The latter does not lend itself that easily to machine interpretation. Formally defined semantics is required for making the language specification (fully) machine readable. The standard and precise definition of modelling language syntax and semantics is indeed pre-requisite for automation. This makes possible e.g. model transformations, interoperability, computer-aided analysis techniques, simulation, and (developing tools for) various other manipulations of models. These are some clear benefits of fixed languages, and one possible strategy to ensure a return on modelling effort.

³This way of addressing meaning is inherent to the functional perspective or action tradition on language. We will elaborate this in the Section 6

The predominant factor for *a priori* fixing the modelling language is therefore the technology (Hoppenbrouwers, 2003), i.e. the fact that mechanical manipulation of models requires fixed representations. It is also assumed that by having standard (and precise) definition of the modelling language, all the meta-discussions on concepts can be avoided. This would contribute to the certainty and efficiency of communication (Hoppenbrouwers, 2003), and shared understanding of models would be easier to reach.

However, as the next two Sections will discuss, standardisation comes at a cost, in particular for pragmatics, which ends up being swept under the carpet.

3 USE OF FIXED LANGUAGES

A key problem in the use of fixed languages in modelling is rooted in the lack of *suitability* of a language for an actual modelling situation. It is indeed often the case that the choice of the modelling language is imposed from the “outside” onto the modelling situation. Typically, the existing modelling infrastructures within the enterprise, the expertise of the modelling team, etc. constrain the choice of the language.

As reported in several surveys on the practice of modelling, see e.g. (Davies et al., 2006) (Anaby-Tavor et al., 2010) (Malavolta et al., 2012), general-purpose modelling languages are the most widely used modelling languages. Nonetheless, these surveys also indicate that ‘variants’ of these languages are in place. For instance, several experience reports of the use of *i** in specific situations (Briand et al., 1995) (Elahi et al., 2008) explain in detail why and how the language was extended to be able to make models that satisfy the needs of the given modelling situation. In the case of ArchiMate, this has e.g. resulted in the suggestion to distinguish between a ‘sketching’ and a ‘designing’ (Lankhorst et al., 2005) variation of the notation (using more sketchy lines and more informal looking fonts). This variation can even be combined in one model to differentiate between the status of different parts of the model. On the same line, (Malavolta et al., 2012) indicate the need for informal ‘variants’ of (software and enterprise) architecture models for their communication to different stakeholders.

In our view, these ‘variants’ are essentially purpose-specific variations of the same original generic modelling language, differing only in their *syntactic and semantic restrictions*, i.e. purpose-specific modelling languages (Bjeković et al., 2012). They emerge from the need to make the language suitable for the communicative task in the actual mod-

elling situation at hand. When the modelling language used is not suitable enough, variations will emerge to compensate for this lack of suitability. Pragmatics re-emerging from under the carpet.

An extreme case of adapting the language to the actual modelling situation is the use of ‘home-grown’ notations (Anaby-Tavor et al., 2010) and/or emergent modelling languages, i.e. the languages that are being constructed along the modelling process. For instance, in (Anaby-Tavor et al., 2010) business architects express a clear preference for home-grown, semi-structured models, since they offer flexibility in terms of re-factoring, delayed commitment to syntax, and closer fit to the inherent way of thinking in these phases. These semi-structured models emerged through the repeated use in similar modelling situations, whereby the sets of concepts and their meanings, and (right level of) restrictions gradually yielded a new language.

Collaborative modelling situations also demonstrate the challenge of adequate modelling support. For instance, in situations whose primary goal is collective knowledge creation, e.g. *developing vision and strategy, scoping the problem, and high-level business design*, the need for simple and intuitive modelling notations, as well as unconstrained medium (e.g. plastic walls, whiteboards) prevails (Bubenko et al., 2010). As most stakeholders do not have modelling expertise, the language and tools have to accommodate this, and thus are required to be simple, intuitive, and corresponding to the natural interaction that occurs in such situations (Barjis, 2009).

Depending on the nature of a modelling situation, the modelling language is, to a greater or lesser degree, able to support the formulation of the desired models. We have discussed a number of different strategies used in practice to compensate for the lack of language suitability. These strategies in one way or the other act on the language specification, aiming to ‘extend’ the actual sets of models which a given language can express. In doing so, there is inevitably the risk of violating the intended pragmatics of the fixed language. However, such a practice may well be an indication that the pragmatic richness of modelling situations to be supported by the language has not been adequately taken into account when the language was designed. In our view, answering this dilemma requires a more fundamental understanding of the role of language in modelling.

4 MODEL INTEGRATION

Since enterprises are modelled using different models/views, it is desirable to maintain their coherence. The use of a wide range of models and modelling languages can easily lead to a fragmentation of the modelling landscape; i.e. a break up of coherence. To avoid or deal with such a situation, different strategies are employed, e.g. (Frank, 2002) (Iacob et al., 2012) (Anaya et al., 2010). We classify them into *language unification* and *language federation* strategy. They both address the integration challenge at the level of fixed language definition. In this section, we discuss some of the challenges that such an approach raises.

The strategy of *unification* of enterprise modelling language(s) has the ambition to define a standard set of constructs in which all the models (i.e. perspectives) can be expressed. The unified language is intended to be used *instead of* the different languages that partially cover the domain of interest. We can observe this logic in the definition of UML, ArchiMate, as well as language of EKD method (Stirna and Persson, 2012). This approach boils down to preventing the fragmentation from occurring in the first place.

The unified language offers a fixed, but integrated, view on some domain of interest. Besides a priori fixing the set of constructs, the standardising effect of the unifying language lies also in the fact that it *a priori* fixes the perspectives for modelling some domain. The integration between the different perspectives, i.e. models, is easier to ensure, given that consistency and coherence rules can be embedded in the language definition. The CASE tool can then *automatically check* these properties.

Regretfully, however, it is nearly impossible to a priori identify which perspectives should be part of an integrated language. The challenge lies in the fact that the relevance of different perspectives (and its related modelling concepts) is highly context-dependent. For instance, different perspectives may be relevant for different (types of) enterprises, or even in different transformation projects of the same enterprise. Moreover, over time, new perspectives may become relevant for a particular enterprise.

To cater for context-dependency, standard modelling languages like UML and ArchiMate offer means for their extension. For instance, UML has the well-known stereotyping mechanism (whose problems are also well-known). In the case of ArchiMate, the very design of the language (Lankhorst et al., 2010) provides different possibilities for extension. However, these mechanisms are of limited scope, they mostly allow for refinement of concepts,

not for adding the entire domains which were not envisaged by the original language definition.

At the same time, one can observe how there is a drive for the ArchiMate language as a *standard* to be extended with additional domains. The move from the ArchiMate 1.0 standard to the ArchiMate 2.0 standard included two additional domains, namely motivation and migration. Further integration between TOGAF and ArchiMate is likely to lead to even more extensions. Moreover, the extensions with e.g. business policies and rules, are also considered (Iacob et al., 2012). *Where will it stop?*

The fact that such unified languages are typically very generic is already an indicator that the perspectives and concepts that are specific to different contexts cannot be covered. Therefore, in the actual use of a unified modelling language in the specific context, the need to *extend* the language with ‘missing domains’ is likely to emerge. On the other hand, defining the comprehensive and overly detailed language covering all the potentially relevant perspectives and related concepts would most likely result in the overly complex language that would be costly to use.

According to (Egyedi, 2007), this tension is inherent to any standard definition process, including modelling standards. The authors argue that defining the context-independent standards (e.g. enterprise- or application-independent, etc.) typically leads to very comprehensive and/or generic standards, therefore also difficult or too expensive to use. Even more, this tension is recognised as a fundamental dilemma in developing standards, which is very difficult to resolve (Egyedi, 2007).

Moreover there is another issue with a unified language approach. A common belief is that by means of *a priori* imposing standardised vocabulary, frequent meta-discussions could be avoided, and the knowledge transfer could be facilitated. However, as well as with perspectives, the languages used in enterprises are also context-dependent, depending on e.g. professional background and education of stakeholders. These factors exert an influence on the default way different people conceptualise (Linden et al., 2012). In that sense, imposing another language, which is ‘outside’ their area of practice, straight from the beginning is likely to cause, and not resolve, conceptual misunderstandings.

A language such as ArchiMate is designed to deal with this issue by enabling users to define their own *viewpoints*, i.e. to have their model (the *view*) derived from the integrated model. However, this viewpoint is still to be defined from the unique fixed ‘footprint’, i.e. unified metamodel.

Conversely to the key drive of *language unification*, the rationale of the strategy of *federating languages* is not to prevent, but rather to allow and manage the variety of modelling languages within the modelling landscapes. To avoid fragmentation, it aims to ensure the links between the modelling languages. A number of different approaches exists to establish these links. We can classify them on several dimensions, for the goal of this discussion.

One classification dimension regards the way in which bridges are constructed between the languages. One approach is to directly connect the languages on a point-to-point basis, e.g. (Bézivin et al., 2006) (Zivkovic et al., 2007) (Fabro and Valduriez, 2009). The links can also be established with the help of a ‘mediator’, as in Unified Enterprise Modelling Language (UEML) (Anaya et al., 2010). The mediator’s role is to serve as the basis to which all the modelling languages are mapped, and to ensure the consistency between the models. In UEML, the unified ‘ontology’ (Opdahl et al., 2012) plays this role. However, it requires the languages to be (re)defined in accordance with the specific grammar, as well as in full formal precision, given that the approach targets the semantic interoperability of tools and associated languages.

We may also distinguish the approaches based on the moment (in the language’s lifetime) when the links are established, i.e. based on the *temporality*. In the case of point-to-point bridges, the links are established between already existing languages. However, the links may also be defined at the moment of language design. In that case, the new language is defined extending or specialising an already existing (more generic) language, the approach akin to the older work on the so-called meta-model hierarchies (Falkenberg and Oei, 1994) (Falkenberg et al., 1998). For instance, (Vallecillo, 2010) discusses in detail different techniques for combining (domain-specific) modelling languages based on this hierarchy logic.

Another dimension of classification may be whether the links are established based on syntactic correspondences only, or whether they also involve semantic correspondences. MDE approaches to model transformations and model weaving (Fabro and Valduriez, 2009) generally only consider the syntactical level. In turn, semantic integration is explored in order to enhance the quality and reduce the complexity of *a priori* establishing language bridges (Karagiannis and Höfferer, 2006). It is however questionable what *kind* of semantics of language constructs can be captured *a priori*. This is especially pronounced for generic languages such as i* (Yu and

Mylopoulos, 1996) or UML, but also relevant for domain-specific languages, as discussed in (Frank, 2011). Without taking the context of language use, the abstract concepts (underlying generic languages) may allow for various interpretations. The precise, i.e. *contextualised* interpretation is, however, necessary for the bridges to be meaningful. For instance, a modelling language such as i* (Yu and Mylopoulos, 1996) can be used for modelling strategic goals of actors in relation to the system, but also to express information systems requirements. In each of these contexts, the *inherent semantics* of e.g. the modelling construct *actor* will vary: when modelling strategic goals of enterprise, *actor* can only be a human actor, while in the context of modelling software requirements, a machine may be an *actor* as well. Without taking this into account, the a priori mappings between i* and ArchiMate (based on language definition and not use) might be meaningless in some of the contexts of i* use.

This context-dependent semantic variation can only be determined by taking into account the context of language use, not *a priori*. Indeed, in discussing the meaning of models, (Thalheim, 2012) distinguishes its two complementary aspects: referential and functional meaning. While the referential meaning is well investigated, the functional meaning relates model elements with the context in which they are used. The key to understanding the functional meaning is in modelling pragmatics.

In addition, as we have seen in the reports of the use of modelling languages, there is an indication that, in various usage contexts, the syntactical variation with respect to the original language specification also takes place. Therefore, the value of building the language bridges *a priori* and based on a priori fixed languages, i.e. out of the context of its actual use, might be questioned.

5 WHAT IS MODELLING?

The remainder of the paper discusses our fundamental understanding of the driving forces and challenges related to modelling and linguistic variety within enterprise modelling landscapes. In our ongoing research, we develop a theory to explain why and how enterprise modelling landscapes emerge and evolve, where we focus on the *use of modelling languages*. Based on such a theory, our ambition is to revisit the integration strategies and propose their realistic variants that better caters for the pragmatics of models/languages. Our view on models and modelling is rooted in semiotics, linguistics and cogni-

tive science. This view is inspired by different related research tackling the fundamental modelling aspects such as (Stachowiak, 1973) (Rothenberg, 1989) (Falkenberg et al., 1998) (Hoppenbrouwers et al., 2005) (Proper et al., 2005) (Thalheim, 2012). We look at the models as being essentially a means of communication about some domain of interest, and the process of modelling as a communication-driven process led by a pragmatic focus (Hoppenbrouwers and Wilmont, 2010).

Though different views on models and modelling exist, as well as many different definitions, we will elaborate the reasons for which we propose the following (general) definition of model (based on (Stachowiak, 1973) (Rothenberg, 1989) (Falkenberg et al., 1998) (Thalheim, 2011)):

*A **model** is an artefact acknowledged by an observer as representing some domain for a particular purpose.*

By stating that a model is an **artefact**, we exclude from the definition the conceptions (Falkenberg et al., 1998) or so-called “mental models” (mental spaces in (Fauconnier, 2010)). Nonetheless, we do consider conceptions as important within the modelling process. Later in this section, we elaborate on their role and importance, especially in the case of collaborative modelling.

With **observer** we refer to the group of people creating (i.e. model creators) and using the model (i.e. model audience). On one extreme, we can refer to the entire society, on the other extreme, to the individual. Though it may not be the general rule, in enterprise modelling context, it is very often the case that model creators are at the same time its audience. The observer is the key element in modelling, as it is only through the appreciation of the observer that some artefact is acknowledged as being the model.

Similarly to (Falkenberg et al., 1998), we define **domain** as any part “part” or “aspect” of the “world” considered relevant by the observer in the given modelling context. The “world” here may refer not only to the “real” world, but also to hypothetical or imagined worlds. Even more, the domain of a model can be another model as well.

A model always has a purpose. This *purposefulness* dimension is explicitly present in most of the model definitions, e.g. (Stachowiak, 1973) (Rothenberg, 1989) (Thalheim, 2011). Although acknowledged as an essential dimension of models, the concept of purpose is rarely defined and its role in the entire modelling process is scantily discussed.

We see the **purpose** of the model as a combination of the following dimensions: (1) the *domain* which the model should pertain to and (2) the intended *us-*

age of the model (e.g. analysis, sketching, contracting, execution, etc.) by its intended *audience*. In line with (Rothenberg, 1989) (Thalheim, 2011), we argue that, although usually implicitly present in modelling, the purpose should be explicit within the modelling process; i.e. the model creator should be aware of the intended usage and audience of the model. This is quite important, since the fitness-for-purpose directly determines/influences the degree to which the models satisfies the conception of the domain within the actual modelling situation. This subsequently influences the value of the model for its intended usage.

As illustrated in Figure 2, when modelling, the observer **O** decides⁴ what the relevant aspects of the “world” under consideration are in the given modelling situation. This results in the *conception of* the domain, c_d . This process of abstracting away from certain aspects of the “world” which are not relevant should be driven by the purpose **p** of the model **m** (Rothenberg, 1989) (Thalheim, 2011) (depicted as influence **1** of the purpose **p** on the relation *conception of*, see Figure 2).

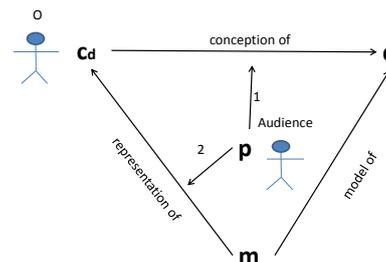


Figure 2: The process of modelling

The observer subsequently tries to shape an artefact (i.e. the model-to-be) in such a way that it adequately represents, for the purpose **p**, his/her conception of the domain c_d . The purpose **p** is a conception as well, i.e. the conception of the purpose of the model-to-be c_p . Even more, the observer **O** also has the conception of the model-to-be, c_m . The modelling process actually consists in the observer’s gradual alignment of these three conceptions (illustrated in Figure 3). This process usually takes place in parallel with the very shaping of the model artefact. When their mutual alignment is achieved, the artefact is acknowledged as the *representation of* the (conception of the) domain **d** for the purpose **p**. In other words, the observer **O** acknowledges that the artefact **m** is a *model of* the domain **d** for the purpose **p**.

⁴Obviously, the observer’s judgement may be influenced by many different factors, e.g. observer’s intentions, experience, previous knowledge, etc. We exclude from our consideration the potential political intentions of the observer.

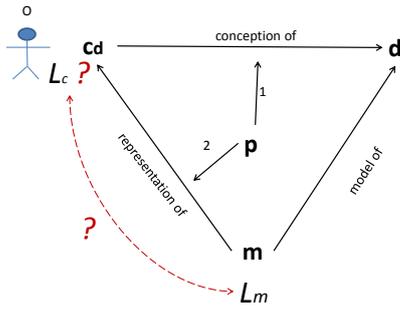


Figure 5: The role of language in the modelling process

elling process, it thus tends to constrain, or at least influence, the conception of a domain. Depending on the actual modelling situation, this pre-conceived filter may prove to be inadequate for the particular observer and for the particular modelling problem.

For instance, research from cognitive linguistics shows that the entire social, cultural, educational and professional background of an observer plays a role in shaping their ‘linguistic personality’ (Novodranova, 2009). This includes their ‘world view’; i.e. their natural way of conceptualising phenomena in the ‘world’ (Schmid, 2010). Likewise, (Linden et al., 2012) shows that different people have different default interpretations of the abstract concepts underlying enterprise modelling languages. It is reasonable to assume that, in a particular modelling situation, (at least) non-modelling-experts form conceptions in a language significantly different from L_m . This is likely to increase the L_c-L_m gap and negatively impact the general suitability of L_m .

Another factor potentially effecting the L_c-L_m gap involves the nature of the modelling task in the particular modelling situation. While the relation between the nature of the modelling problem and the suitability of the modelling language needs further study, the empirical data indicates rather negative influence of overly restrictive (in terms of syntactic-semantic restrictions embedded in) fixed language on the creativity in modelling. For instance, (Anaby-Tavor et al., 2010) observe the inadequacy of fixed modelling languages to support the exploration phases where things are unclear and ambiguous, and models are used to organise information, gain insight, envision alternative possible futures etc. (Anaby-Tavor et al., 2010). Similarly, (Bubenko et al., 2010) observe that in highly creative and collaborative situations such as vision and strategy development, rather informal and intuitive notations (and mediums) seem to be of better support.

In its role as a *medium*, a modelling language should accommodate the formulation of models, while allowing their mechanical manipulation. The

potential added value of the modelling language, from this perspective, lies primarily in its re-usability across different modelling problems, and the extent to which the language specification is machine readable.

As discussed in Section 2, the reusability of a language relates to its expressiveness. Obviously, this makes sense for the development of tools and automated model management. However, while general-purpose modelling languages are usually more expressive, they are less suitable than domain or purpose-specific languages for specific problem domains and modelling situations. These languages incorporate in their definition concepts that are tuned to the modelling of particular domains. The overall aim is to foster modelling productivity, facilitate the understanding of the models by the domain stakeholders and increase the overall quality of resulting models, in particular semantic and pragmatic quality (Krogstie et al., 2006). While domain and purpose-specific languages seem to correspond to the natural, i.e. human, need for suitability of the modelling *language*, they are not easy to reuse across different situations. General-purpose modelling languages, on the other hand, are easier to reuse for modelling different domains. Nevertheless, the interpretation challenge of the models expressed in them is more pronounced.

It certainly does not make sense to have situation-specific modelling language emerge from scratch in each new modelling situation. However, it does make sense to embed in the (generic) meta-model the elements that are repeatedly discussed in similar modelling contexts. Therefore, there is a need to carefully balance potential sets of models one would like to express in a language, and potential sets of modelling situations one would like to support.

The second added value of a language as a medium is machine readability. This is driven by the need for automated manipulation of models. This is achieved by formal, i.e. precise and unambiguous, definition of both the syntax and semantics of the language, usually in a mathematical language. This essentially boils down to expressing the semantics of the model/language in terms of another syntactic representation, i.e. expressing the *syntactic semantics*. Though necessary for the machine’s correct interpretation of the model, this kind of a priori fixed semantics does not tell anything about what the model means to the observer. In particular, it allows by no means to a priori precisely capture the meaning of a model as it occurs in the actual modelling situation, i.e. the model’s *pragmatic semantics*.

6.2 Language Variety

We are now able to suggest two primary drivers for the variety within enterprise modelling landscapes. In our view, these drivers relate to:

Abstraction variety – Abstraction is at the heart of modelling. As we have seen, it boils down to *purposefully* neglecting irrelevant details of the observed “world”. The need for differing levels of abstraction in dealing with an enterprise is related to its complexity/multifaceted-ness. The abstraction variety thus leads to the increase in number of needed perspectives, i.e. models, in modelling.

Manifestation variety – The manifestation here refers to the way the model is *represented* on the *medium*. Fundamentally, the need for this kind of variety is rooted in the complexity and heterogeneity of social structures underlying modelling. The more heterogeneity in stakeholder groups, the more likely is the need for different manifestations of arguably the “same” model. This invites an increase in linguistic variety on top of the abstraction variety.

These types of variety are illustrated in Figure 6. Throughout the paper, we have seen that these factors are to a large extent situational, i.e. they depend on the particular enterprise and enterprise modelling effort, involved stakeholders (observer), the purpose for which models are created in this effort, etc. This leads us to the conclusion that modelling landscapes should be *situated*. Indeed, the ‘standard eroding’ effects discussed in Sections 3 and 4, might be seen as the manifestation of the need to make these landscapes better situated, driven by the *pragmatic needs* of the wider organisational context that the landscapes cover.

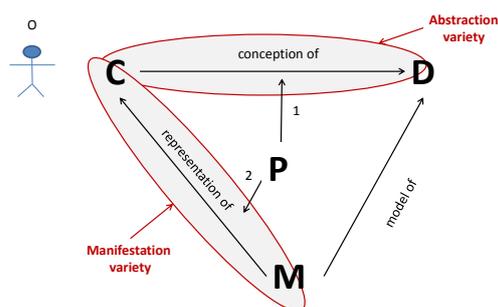


Figure 6: Sources of variety in the modelling landscapes

Evidently, both of these drivers stem from the *language role* of the modelling language. They also fundamentally conflict with the drivers for the a priori standardisation of the language, which stem from its *medium role*. We believe that this natural polarity deserves careful management, rather than denial. It should by no means be swept under the carpet.

7 CONCLUSION & OUTLOOK

Based on the discussions so far, we posit that, at the heart of the challenge of creating integrated modelling landscapes, lies the question: *What can be a priori fixed in a modelling language?* To create more balanced strategies that cater for the pragmatic needs of modelling landscapes, it is necessary to carefully examine which aspects may be feasible to a priori fix in the modelling language. In this final section, and relying on the presented fundamental view on modelling and language, we draw some initial conclusions as a tentative (though partial) answer to this question.

First of all, the semantics. Given that (the conception of) the *domain* actually does not ‘exist’⁵ a priori but emerges in the very process of its modelling, the semantics, in the *pragmatic semantics* sense, also cannot be captured a priori. One would expect to be able to at least fix the grammar (abstract syntax) and symbols used for its representation on the medium (concrete syntax). To the extent to which the grammar is tuned to the needs of the intended set of modelling situations, it can be *a priori* fixed. One can also start by only a priori fixing the core grammar, and allowing its further refinement *a posteriori* during the use of the language. For instance, it can be possible to start the modelling process with lightly constrained vocabulary adapted to the domain/purpose (e.g. based on some historical heuristics), and then gradually include more formalisation, to the extent necessary for the intended usage and audience of the model. Needless to say, this would necessitate modelling infrastructures to be more flexible. A growing interest in this subject can indeed be observed, e.g. (Cho et al., 2011) (Ossher et al., 2009).

The point we aim to make is that, although having an a priori fixed representation to a large extent facilitates the development of tools and automation of model manipulations, fixing the language that is to be used in human communication may seriously damage its capacity to adequately express thoughts, i.e. conceptions of domains in the given modelling situation. Even if carefully defined, the standardised enterprise modelling language will inevitably demonstrate the need to evolve, to adapt to the dynamically changing

⁵The term *exist* is used here in the sense of Heidegger’s notion of *breaking down*, discussed in (Winograd and Flores, 1986). Indeed, “Heidegger insists that it is meaningless to talk about the existence of objects and their properties in the absence of concerned activity, with its potential for breaking down. What really is is not defined by an objective omniscient observer, nor is it defined by an individual – the writer or computer designer – but rather by a space of potential for human concern and action” (Winograd and Flores, 1986, p.37).

‘reality’ of enterprises and their environments, and thus to the human sense-making of that ‘reality’.

As our next step, we aim to further explore the theoretical and practical considerations presented in this paper. We aim to start by analysing the available instruments for modelling language design, adaptation and combination, and the potential of their improving or combining in order to support purpose-specific language adaptations. In addition, we aim to extend these instruments to allow for explicit modelling of the modelling pragmatics. For instance, megamodel (Favre and Nguyen, 2004), viewpoint (ISO, 2011), metamodel hierarchies (Falkenberg et al., 1998), metamodel inference (Ossher et al., 2009) etc. are some of the instruments of our particular interest.

ACKNOWLEDGEMENTS

This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (www.fnrl.lu), via the PEARL programme.

REFERENCES

- Anaby-Tavor, A., Amid, D., Fisher, A., Bercovici, A., Ossher, H., Callery, M., Desmond, M., Krasikov, S., and Simmonds, I. (2010). Insights into enterprise conceptual modeling. *Data Knowl. Eng.*, 69(12):1302–1318.
- Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevicius, R., Opdahl, A. L., Panetto, H., and Verdecho, M. (2010). The Unified Enterprise Modelling Language - Overview and Further Work. *Computers in Industry*, 61:99–111.
- Barjis, J. (2009). Collaborative, Participative and Interactive Enterprise Modeling. In *ICEIS*, pages 651–662.
- Bézivin, J., Bouzitouna, S., Fabro, M. D. D., Gervais, M.-P., Jouault, F., Kolovos, D. S., Kurtev, I., and Paige, R. F. (2006). A canonical scheme for model composition. In *ECMDA-FA*, pages 346–360.
- Bjeković, M., Proper, H. A., and Sottet, J.-S. (2012). Towards a coherent enterprise modelling landscape. In *PoEM (Short Papers)*. CEUR-WS.org.
- Briand, L. C., Melo, W. L., Seaman, C. B., and Basili, V. R. (1995). Characterizing and Assessing a Large-Scale Software Maintenance Organization. In *ICSE*, pages 133–143.
- Bubenko, J. A. j., Persson, A., and Stirna, J. (2010). An Intentional Perspective on Enterprise Modeling. In Salinesi, C., Nurcan, S., Souveyet, C., and Ralyté, J., editors, *Intentional Perspectives on Information Systems Engineering*, pages 215–237. Springer.
- Cho, H., Sun, Y., Gray, J., and White, J. (2011). Key challenges for modeling language creation by demonstration. In *ICSE 2011 Workshop on Flexible Modeling Tools, FLEXITOOLS*.
- Clark, H. (1993). *Arenas of Language Use*. University of Chicago Press.
- Cruse, A. (2011). *Meaning in Language: An Introduction to Semantics and Pragmatics*. Oxford Textbooks in Linguistics. Oxford University Press.
- Davies, I., Green, P. F., Rosemann, M., Indulska, M., and Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data Knowl. Eng.*, 58(3):358–380.
- Delen, D., Dalal, N. P., and Benjamin, P. C. (2005). Integrated modeling: the key to holistic understanding of the enterprise. *Commun. ACM*, 48(4):107–112.
- Egyedi, T. M. (2007). Standard-compliant, but incompatible?! *Computer Standards & Interfaces*, 29(6):605–613.
- Elahi, G., Yu, E. S. K., and Annosi, M. C. (2008). Modeling Knowledge Transfer in a Software Maintenance Organization - An Experience Report and Critical Analysis. In *PoEM 2008*, pages 15–29.
- Fabro, M. D. D. and Valduriez, P. (2009). Towards the efficient development of model transformations using model weaving and matching transformations. *Software and System Modeling*, 8(3):305–324.
- Falkenberg, E. and Oei, J. (1994). Meta Model Hierarchies from an Object–Role Modelling Perspective. In *Proceedings of the First International Conference on Object–Role Modelling (ORM–1)*, pages 218–227.
- Falkenberg, E. D., Hesse, W., Lindgreen, P., Nilsson, B. E., Oei, J., Rolland, C., Stamper, R. K., Assche, F. J. V., Verrijn-Stuart, A. A., and Voss, K. (1998). FRISCO - A Framework of Information System Concepts - The FRISCO Report. Technical report, IFIP WG 8.1 Task Group FRISCO.
- Fauconnier, G. (2010). Mental Spaces. In Geeraerts, D. and Cuyckens, H., editor, *The Oxford Handbook of Cognitive Linguistics*, pages 351–376. Oxford University Press.
- Favre, J. and Nguyen, T. (2004). Towards a megamodel to model software evolution through transformation. In *SETRA workshop*.
- Frank, U. (2002). Multi-perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In *Proceedings of HICSS’02*, volume 3. IEEE.
- Frank, U. (2011). Some Guidelines for the Conception of Domain-Specific Modelling Languages. In *EMISA*, pages 93–106.
- Gordijn, J. and Akkermans, H. (2003). Value based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134.
- Greefhorst, D., Koning, H., and Van Vliet, H. (2006). The many faces of architectural descriptions. *Information Systems Frontiers*, 8(2):103–113.
- Harel, D. and Rumpe, B. (2004). Meaningful Modeling: What’s the Semantics of “Semantics”? *IEEE Computer*, 37(10):64–72.
- Hoppenbrouwers, S. (2003). *Freezing Language; Conceptualisation processes in ICT supported organisations*.

- PhD thesis, University of Nijmegen, Nijmegen, The Netherlands.
- Hoppenbrouwers, S., Proper, H., and Weide, T. v. d. (2005). A Fundamental View on the Process of Conceptual Modeling. In *Conceptual Modeling - ER 2005*, volume 3716, pages 128–143. Springer.
- Hoppenbrouwers, S. J. B. A. and Wilmont, I. (2010). Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In *PoEM 2010*, pages 190–204.
- Iacob, M.-E., Jonkers, H., Lankhorst, M., Proper, H., and Quartel, D. (2012). *ArchiMate 2.0 Specification*. The Open Group.
- ISO (2011). *ISO/IEC/IEEE 40210:2011 Systems and software engineering – Architecture description*.
- Kaidalova, J., Seigerroth, U., Kaczmarek, T., and Shilov, N. (2012). Practical Challenges of Enterprise Modeling in the Light of Business and IT Alignment. In *PoEM 2012*, pages 31–45.
- Karagiannis, D. and Höfner, P. (2006). Metamodels in Action: An overview. In *ICSOFT (1)*. INSTICC Press.
- Karlsen, A. (2011). Enterprise Modeling Practice in ICT-Enabled Process Change. In *PoEM 2011*, pages 208–222.
- Krogstie, J., Sindre, G., and Jørgensen, H. D. (2006). Process models representing knowledge for action: a revised quality framework. *EJIS*, 15(1):91–102.
- Lankhorst, M., editor (2005). *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Germany.
- Lankhorst, M., Torre, L. v. d., Proper, H., Arbab, F., and Steen, M. (2005). Viewpoints and Visualisation. In (Lankhorst, 2005), pages 147–190.
- Lankhorst, M. M., Proper, H. A., and Jonkers, H. (2010). The Anatomy of the ArchiMate Language. *IJISMD*, 1(1):1–32.
- Linden, D. J. T. v. d., Hoppenbrouwers, S., Lartseva, A., and Molnar, W. (2012). Using Psychometrics to Explicate Personal Ontologies in Enterprise Modeling. *Applied Ontology*, 7:1–15.
- Malavolta, I., Lago, P., Muccini, H., Pellicone, P., and Tang, A. (2012). What industry needs from architectural languages: an industrial study. Technical report, University of L'Aquila.
- Moody, D. (2009). The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779.
- Novodranova, V. F. (2009). Representation of the Language Personality in LSP. In *Reconceptualizing LSP, Online proceedings of the XVII European LSP Symposium*.
- OMG (2003). UML 2.0 Superstructure Specification – Final Adopted Specification. Technical Report ptc/03–08–02.
- OMG (2006). Semantics of Business Vocabulary and Rules (SBVR). Technical Report dtc/06–03–02, Object Management Group, Needham, Massachusetts.
- OMG (2008). Business Process Modeling Notation, V1.1. OMG Available Specification OMG Document Number: formal/2008-01-17.
- Opdahl, A., Berio, G., Harzallah, M., and Matulevicius, R. (2012). Ontology for Enterprise and Information Systems Modelling. *Applied Ontology*, 7(1):49–92.
- Ossher, H., Bellamy, R. K. E., Amid, D., Anaby-Tavor, A., Callery, M., Desmond, M., de Vries, J., Fisher, A., Fraunhofer, T., Krasikov, S., Simmonds, I., and Swart, C. (2009). Business insight toolkit: Flexible pre-requirements modeling. In *ICSE Companion*, pages 423–424. IEEE.
- Proper, H. A., Verrijn-Stuart, A. A., and Hoppenbrouwers, S. (2005). On Utility-based Selection of Architecture-Modelling Concepts. In *APCCM 2005*, pages 25–34.
- Rothenberg, J. (1989). The Nature of Modeling. In *Artificial intelligence, simulation & modeling*, pages 75–92. John Wiley & Sons, Inc., USA.
- Schmid, H.-J. (2010). Entrenchment, Salience, and Basic Levels. In Geeraerts, D. and Cuyckens, H., editor, *The Oxford Handbook of Cognitive Linguistics*, pages 117–138. Oxford University Press.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Springer, Berlin, Germany.
- Stirna, J. and Persson, A. (2012). Evolution of an Enterprise Modeling Method - Next Generation Improvements of EKD. In *PoEM 2012*, pages 1–15.
- Thalheim, B. (2011). The Theory of Conceptual Models, the Theory of Conceptual Modelling and Foundations of Conceptual Modelling. In *Handbook of Conceptual Modeling*, pages 543–577. Springer.
- Thalheim, B. (2012). Syntax, Semantics and Pragmatics of Conceptual Modelling. In *NLDB*, pages 1–10. Springer.
- Vallecillo, A. (2010). On the Combination of Domain Specific Modeling Languages. In *ECMFA*, pages 305–320.
- Vernadat, F. (2002). UEML: Towards a Unified Enterprise Modelling Language. *International Journal of Production Research*, 40(17):4309–4321.
- Wagter, R., Proper, H., and Witte, D. (2012). A Practice-Based Framework for Enterprise Coherence. In *Proceedings of PRET 2012*, volume 120 of *LNBIP*. Springer. To appear.
- Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition - A New Foundation for Design*. Ablex Publishing Corporation.
- Winter, R. and Fischer, R. (2007). Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. *Journal Of Enterprise Architecture*, 3(2):7–18.
- Wood-Harper, A., Antill, L., and Avison, D. (1985). *Information Systems Definition: The Multiview Approach*. Blackwell, Oxford, United Kingdom.
- Yu, E. and Mylopoulos, J. (1996). Using goals, rules, and methods to support reasoning in business process reengineering. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 5(1):1–13.
- Zachman, J. (1987). A framework for information systems architecture. *IBM Systems Journal*, 26(3).
- Zivkovic, S., Kühn, H., and Karagiannis, D. (2007). Facilitate Modelling Using Method Integration: An Approach Using Mappings and Integration Rules. In *ECIS*, pages 2038–2049.