Ontology-Based Evaluation and Design of Domain-Specific Visual Modeling Languages

Giancarlo Guizzardi, Luís Ferreira Pires and Marten van Sinderen

CTIT, University of Twente, Enschede, The Netherlands
guizzard@cs.utwente.nl, pires@cs.utwente.nl, sinderen@ctit.utwente.nl

Introduction

In recent years, increasing attention has been paid to the development of *domain-specific visual modeling languages* (DSVLs). It is believed that these languages can lead to an increase in productivity in the modeling activity and contribute to the production of models that are more flexible, reusable and easier to maintain than models produced by using general-purpose modeling languages (Tolvanen et al, 2004). However, in order to be effective, a DSVL must be defined taking into account the needs of its client users. From their perspective, the use of the language should be satisfactory in the following terms: (i) it should easy for a user of the language to communicate, understand and reason with the produced models (*comprehensibility appropriateness*); (ii) The language should be truthful to the domain in reality that it is supposed to represent (*domain appropriateness*).

In this article, we present an ontology-based method for the evaluation and (re)design of DSVLs that reinforces properties (i) and (ii) above. We start by presenting the different elements of languages design, namely, *syntax*, *semantics* and *pragmatics*. After that, we discuss the subject of *formal ontologies* and its relation to each of these elements and present the proposed method. Finally, we illustrate this method with the design of a visual modeling language in the domain of genealogy.

Elements of Language Design

According to (Morris, 1938) a language comprises three parts: *syntax*, *semantics* and *pragmatics*. Syntax is devoted to *"the formal relation of signs to one another"*. In order to communicate, agents must agree on a common communication language. This fixes the sets of signs that can be exchanged (syntax) and how these signs can be combined in order to form valid expressions in the language (syntactical rules). The set of available modeling primitives of a language forms the lexical layer and the language _abstract syntax_ (typically defined in terms of a _metamodel_) delimits the set of grammatically correct models that can be constructed using that language.

A syntactic item is essential to give a concrete and persistent status to some information, but it is in itself, however, vacuous in terms of meaning. Therefore, participants in a communication process must also share the same meaning for the syntactical constructs being communicated, i.e., they must interpret in a compatible way the expressions of the communication language being used. Thus, to assign meaning to a syntactic sign, a mapping is necessary, between that sign and some entity in reality that it represents. The Semantics of a given syntactic item can then be defined as *"the relation of signs to real world entities they represent"* (Morris, ibid.).

Whilst the abstract syntax defines the rules for the creation of well-formed sentences of a given language, the vocabulary, or _concrete syntax_, provides a concrete representational system for expressing the elements of a given the domain. In sentential languages, there is a clear separation between vocabulary, syntax and semantics. The same does not hold for visual languages. For example, a visual vocabulary may include shapes such as circles, squares, arcs and arrows, all of differing sizes and colors. These objects often fall naturally into a hierarchical typing which almost certainly constrains the syntax and, furthermore, informs the semantics of the system (Gurr, 1999). This idea is illustrated by Figure 1 below, in which two different languages are used to express logical syllogisms. The sentential language of Figure 1.a and the graphical language of Figure 1.b are semantically equivalent. Despite that, the inference step that culminates with conclusion (iii) is performed in a much more straightforward way in the language of Euler's circles (Figure 1.b).

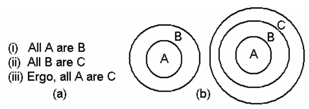


Fig. 1. Logical Syllogism represented in (a) a sentential language and (b) in the visual language of Euler's Circles.

This classic example shows how semantic information can be directly captured in a visual symbol. Here a sequence of valid operations is performed which cause some consequence to become manifest in a diagram, where that consequence is not explicitly insisted upon by the operations. This is because that the *partial order* properties of the set inclusion relation are represented via the similarly transitive, irreflexive and asymmetric visual properties of proper spatial inclusion in the plane, i.e. the representing relation has the same semantic properties of the represented relation.

In visual languages, intrinsic properties of the representation system can be systematically used to directly correspond to properties in the represented domain. This can lead to major increases in the effectiveness for performing specific tasks of the diagrams produced using this language (Gurr, ibid.). The benefits that can be achieved by exploring the inherent properties of representation systems (as well as the potential traps of ignoring them) are derived from the relation between a representation system of visual syntax and the human users interpreting that representation. Thus, following (Morris, ibid.), if syntax refers to "*the formal relation of signs to one another*", and semantics to *"the relation of signs to real world entities they represent"*, then pragmatics refers to *"the relation of signs to (human) interpreters"*.

Ontology, (Meta)Conceptualization and Language

One of the main success factors behind the use of a modeling language is its ability to provide to its target users a set of modeling primitives that can directly express relevant domain abstractions. Domain abstractions are constructed in terms of concepts, i.e., abstract representations of certain aspects of entities that exist in a given domain that we name here *a domain conceptualization.* An abstraction of a certain *state of affairs* expressed in terms of a set of domain concepts, i.e., according to a certain conceptualization, is termed a *domain abstraction* in this work. Domain abstractions and conceptualizations are intangible entities that only exist in the mind of the user or a community of users of a language. In order to be documented, communicated and analyzed they must be captured, i.e. represented in terms of some concrete artifact. This implies that a language is necessary for representing them in a concise, complete and unambiguous way. The relations between these entities are elaborated in Figure 2, which depicts the distinction between a domain abstraction and its representation, and their relationship with the domain conceptualization and the representation language. In the scope of this work the representation of a domain abstraction in terms of a representation language $\mathcal{L}$ is called a *model or specification* and the language $\mathcal{L}$ used on its creation is called a modeling (or specification) language.

The position defended here is that a particular model $\mathcal{M}$ (produced in a modeling language $\mathcal{L}$) is considered a adequate model of a domain abstraction $\mathcal{A}$ if it preserves the structure of $\mathcal{A}$. Likewise, we can say that a modeling language $\mathcal{L}$ is appropriate to model a domain $\mathcal{D}$ according to a conceptualization $\mathcal{C}$ of $\mathcal{D}$ if $\mathcal{L}$ allows model designers to build models $\mathcal{M}$ which preserve the structure of the domain abstractions articulated with $\mathcal{C}$. We then advocate that the adequacy of a modeling language to represent phenomena in a given domain can be systematically evaluated by comparing, on one hand, a concrete representation of the worldview underlying that language (captured by that language's *metamodel*) to, on the other hand, a concrete representation of a domain conceptualization, or a *domain ontology*. The truthfulness to reality (*domain appropriateness)* and conceptual clarity *(comprehensibility appropriateness)* of a modeling language depend on the level of homomorphism between these two entities. The stronger the match between an abstraction in reality and its representing model, the easier is to communicate and reason with that model.

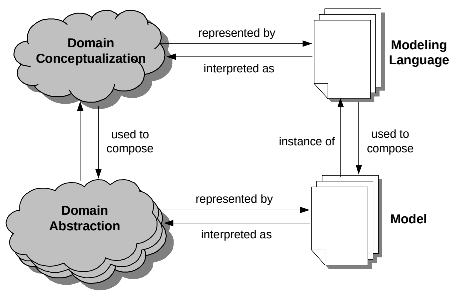


Fig 2. Relation between conceptualization, abstraction, language and model.

In Guizzardi et al. (2005), we have discussed a number of properties that should be reinforced for an isomorphic mapping to take place between an ontology $\mathcal{O}$ representing a domain $\mathcal{D}$ and a domain language's metamodel. These properties are briefly discussed in the sequel: (a) *Soundness:* A language $\mathcal{L}$ is *sound* w.r.t. to a domain $\mathcal{D}$ iff every modelling primitive in the language has an interpretation in terms of a domain concept in the ontology $\mathcal{O}$; (b) *Completeness:* A language $\mathcal{L}$ is *complete* w.r.t. to a domain $\mathcal{D}$ iff every concept in the ontology $\mathcal{O}$ of that domain is represented in a modelling primitive of that language; (c) *Lucidity:* A language $\mathcal{L}$ is *lucid* w.r.t. to a domain $\mathcal{D}$ iff every modelling primitive in the language represents at most one domain concept in $\mathcal{O}$. (d) *Laconicity:* A language $\mathcal{L}$ is *laconic* w.r.t. to a domain $\mathcal{D}$ iff every concept in the ontology $\mathcal{O}$ of that domain is represented at most once in the metamodel of that language.

Unsoundness, *Non-Lucidity*, *Non-Laconicity* and *Incompleteness* violate what the philosopher of language H.P.Grice (1975) names *conversational* *maxims* that states that a speaker is assumed to make contributions in a dialogue which are *relevant*, *clear*, *unambiguous*, and *brief*, *not overly informative* and *true according to the speaker's knowledge*. Whenever models do not adhere to these conversational maxims, *they* can communicate incorrect information and induce the user to make incorrect inferences about the semantics of the domain.

In regards to the property of completeness, when mapping the elements of a domain ontology to a language metamodel we must guarantee that these elements are represented in their full formal descriptions. In other words, the metamodel $\mathcal{MT}$ of language $\mathcal{L}$ representing the domain ontology $\mathcal{O}$ must also represent this ontology's full axiomatization. In formal, model-theoretic terms, this means that these entities should have the same set of logical models. In Guizzardi et al. (2005), we discuss this topic in depth and present a formal treatment of this idea. The set of logical models of $\mathcal{O}$ represent the state of affairs in reality deemed possible by a given domain conceptualization. In contrast, the set of logical models of $\mathcal{MT}$ stand for the world structures which can be represented by the grammatically correct specifications of language $\mathcal{L}$. In summary, we can state that if a domain ontology $\mathcal{O}$ is fully represented in a language metamodel $\mathcal{MT}$ of $\mathcal{L}$, then the only grammatically correct models of $\mathcal{L}$ are those which represent state of affairs in reality deemed possible by the domain conceptualization represented by $\mathcal{O}$.

By representing a domain conceptualization in terms of a concrete artefact (a domain ontology) we can systematically evaluate the adequacy of existing modeling languages to represent phenomena in that domain. However, we can also use the domain ontology as the starting point for the design of new modeling language in that domain. In both cases, the objective is to reach a language metamodel which is isomorphic to an ontology representing a conceptualization of that domain. When this isomorphism is guaranteed, some pragmatic benefits are already achieved. However, additional benefits are achieved when another isomorphism is reinforced, namely, between the language metamodel (describing the language's abstract syntax) and the system of representations that forms the concrete syntax of the language.

Take, for instance, the model depicted in Figure 3. As one can notice, the models of Figures 3.a and 3.b are isomorphic. In this figure the arrow symbol represents the subsumption relation between types. The dashed arrow symbol, in particular, represents a subsumption relation between two different modes of type, namely, a *kind* (e.g., City) and a *phase* type (e.g., Town) (Guizzardi et al., 2004). Phases represent types which are instantiated by an individual only *contingently* (in the modal sense). A kind, in contrast, is instantiated by its instances *necessarily* (again, in the modal sense). For instance, every instance of the type person must be a person, i.e., it cannot cease to be a person without ceasing to exist. However, a person can be a student in a world w and cease to be so in world w' without ceasing to be the same individual (the same person). In this example, the same city *a* can be considered a town in a world w and a metropolis in w', but still maintaining its cross-world identity. Likewise, in Figure 3.b, the color property of a geometric figure is considered one of its contingent properties. Thus, a particular circular form is assumed to be able to change its color while maintaining a continuous visual percept.

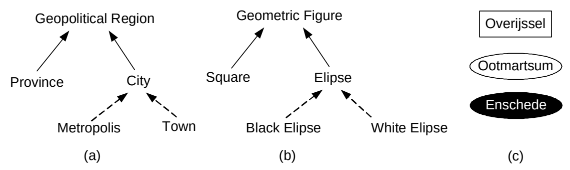


Fig 3. Examples of (a) a illustrative domain ontology/metamodel; (b) a system of visual syntax isomorphic to the metamodel in (a); (c) a particular valid model according to (a).

This example highlights some important aspects of the approach discussed here. To start with, since ontologies are themselves models, they must also be represented in a certain modeling language. Take for instance, the small geopolitical regions ontology depicted above (Figure 3.a). In terms of Figure 2, this ontology can be considered a representation of a geopolitical regions conceptualization (upper-left corner), and isomorphic to the metamodel of language for describing geopolitical regions (upper-right corner). A possible model in the language defined by this metamodel (bottom-right corner) is the one of figure 3.c. This model represents a state of affairs (bottom-left corner) in which there exists a province named Overijssel, a city named Enschede, which is considered a metropolis, and a city named Ootmartsum, which is considered a town.

However, if we put the ontology of Figure 3.a in the bottom-right corner of Figure 2, then it can be considered as a valid model of a *general ontology representation language* (upper-right corner), containing primitives such as the (dashed) arrows in Figure 3.a. In this case, what should be the domain-independent meta-conceptualization that this general ontology representation language should commit to? We argue that it should be a system of general categories and their ties, which can be used to articulate domain-specific common sense theories of reality. This meta-conceptualization should comprise a number of domain-independent theories (e.g., theory of parts and wholes, types and subsumption, identity, existential dependence, etc.), which are able to characterize aspects of real-world entities irrespective of their particular nature. The development of such general theories of reality is the business of the philosophical discipline of *Formal Ontology*. A concrete artefact representing this meta-conceptualization is named a *Foundational Ontology*.

In a series of papers (e.g., Guizzardi et al., 2004, 2005), we have employed the evaluation method discussed in this section to evaluate and re-design the Unified Modeling Language (UML) for the purpose of conceptual modelling and ontology representation. Following this method, we compare the UML 2.0 metamodel with a philosophically principled foundational ontology. As a result we were able not only to propose a well-founded modeling profile that extends the language, but also to provide real-world semantics for the elements that constitute this profile.

The method proposed here evaluates the quality of domain-specific modelling language w.r.t. a domain ontology. As a consequence, the quality of a modelling language strongly depends on the quality of the ontology in which it is based. For this reason, the use of *suitable* ontology representation language to create these domain ontologies plays an important in this approach. Moreover, as illustrated in Figure 3, a representation language which recognizes more subtle distinctions among domain concept categories, allows for the construction of visual syntaxes which are also sensitive to these distinctions, or as defended by Gurr (1999): the more we know about a domain being represented, the bigger the chance we have of devising pragmatically efficient languages for that domain.

In summary, the approach proposed here for domain-specific visual language evaluation and re-design comprises of: (1) a number of properties that should be reinforced for guaranteeing the isomorphism between: (i) a domain ontology (representing the target domain conceptualization) and a metamodel of the language; (ii) a metamodel of the language (defining its set of valid models) and one representing its system of concrete visual syntax; (2) a general ontology representation language that can be used to construct these domain ontologies.

Example: A DSVL for the Genealogy Domain

In the sequel we illustrate the notions discussed so far with an example in the domain of genealogical relations. A certain conceptualization of this domain can be articulated by considering concepts such as *person*, *living person*, *deceased person*, *father*, *mother*, *offspring*, *fatherOf* and *motherOf*. These concepts are related to each other and have their interpretation constrained by axioms imposed on their definitions. Figure 4 depicts a domain ontology representing a possible conceptualization in this domain. The language used to represent this model is the ontologically well-founded version of UML proposed in (Guizzardi et al., 2004). The diagram on this picture can be complemented by the following axioms: (a) A person *x* is a *parentOf* person *y* iff *x* is *fatherOf y* or *x* is *motherOf y*; (b) A person *x* is a *ancestorOf* person *y* iff *x* is *parentOf y* or there is a person *z* such that *z* is an *parentOf y* and *x* is *ancestorOf z*; (c) A person cannot be its own ancestor; (d) If a person *x* is an *ancestorOf* person *y* then *y* cannot be an *ancestorOf x*; (d) If a person *x* is an *ancestorOf* person *y* and *y* is an *ancestorOf* person *z* then *x* is an *ancestorOf z*.

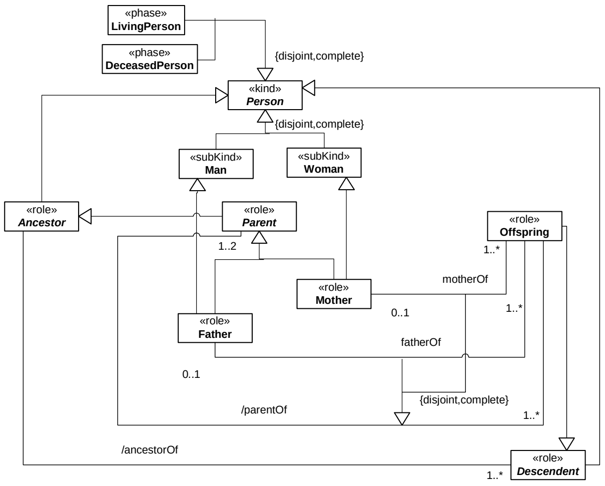


Figure 4. An ontology for the genealogy domain**.**

By representing a conceptualization of this domain in terms of this concrete artefact we can design a language to express phenomena on this domain capturing characteristics that this conceptualization deems relevant. For instance, according to this domain ontology, Person is an abstract type, i.e., one that cannot have direct instances. This type is partitioned in two independent suptyping structures:

– **<u>Man, Woman:</u>** this partition represents that every individual person (instance of type Person) in the universe of discourse is either a man (an instance of Man) or woman (instance of Woman). Moreover, due to the «subKind» stereotype, it states that every man is *necessarily* a man (in the modal sense). Analogously, the same applies to instances of the type Woman;

– **LivingPerson, DeceasedPerson:** this partition represents that every individual person in the universe of discourse is either a living person or a deceased one. However, in contrast to the ⟨Man,Woman⟩ partition, an instance of LivingPerson *is not necessarily so* (in the modal sense). That is to say that for every x such that x is LivingPerson there is a counterfactual situation in which x is not a LivingPerson, which in this case, implies that x is a DeceasedPerson in this counterfactual situation. Analogously, the same applies to instances of DeceasedPerson. These facts are implied by the presence of «phase» stereotyped constructs and the associated constraint in the modelling profile used that phases must be defined in a partition (Guizzardi et al., 2004).

A cross-relation of these two partitions give us four concrete types, i.e., types that can have direct instances, let us name them *LivingMan*, *DeceasedMan*, *LivingWoman* and *DeceasedWoman*. Every instance of person in a given situation is necessarily an instance of one of these types. A suitable modeling language in this domain must have modelling primitives that conform to these constraints. Other constraints on the possible models according to this ontology include: that every *offspring* can have at maximum one *father* and one *mother*; the *ancestorOf* relation (defined to hold between instances of *Person*) is irreflexive, antisymmetric and antitransitive. The set of constraints captured in this ontology must be taken into account in the design (an evaluation) of a language to model genealogical relations.

By having a conceptualization (abstract entity) represented in terms of domain ontology and, by applying the method discussed in section 3, one can, in a precise manner, design a suitable modeling language for that given domain. In this particular case, we are able to design a language $\mathcal{L}_1$ which structure is isomorphic to the ontology of Figure 4. The primitives of this language are presented in Figure 5.

The following characteristics of language $\mathcal{L}_1$ should be observed: (i) the language contain modeling primitives that represent all concrete classifiers (*LivingMan*, *DeceasedMan*, *LivingWoman*, *DeceasedWoman)* and non-derived relations present in the ontology (*fatherOf, motherOf)*. Consequently, we can say that language is expressive enough to model all characteristics of the domain that are considered relevant by the underlying ontology; (ii) there is no case of construct redundancy, construct overload or construct excess; (iii) since the metamodel of this language is identical to the ontology of Figure 4, the well-formedness rules in this metamodel also includes the axioms of that ontology. In other words, the only syntatically valid models in language $\mathcal{L}_1$ are those that represent state of affairs deemed admissible by that ontology. As a consequence, the models depicted in Figure 6 (a-c) are not syntactically valid in $\mathcal{L}_1$. A valid model in this language is depicted in Figure 6.d.

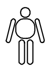
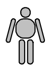
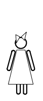
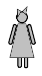
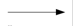
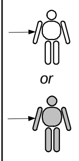
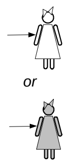
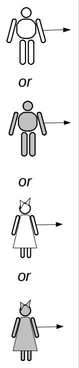
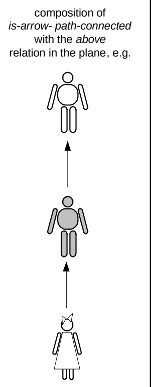
Language					 *(is-arrow-directly-connected)*				
Ontology	Living Man	Deceased Man	Living Woman	Deceased Woman	Parent of	Father	Mother	Offspring	Ancestor of

Fig 5. Domain Concepts and their representing modeling primitives in $\mathcal{L}_1$.

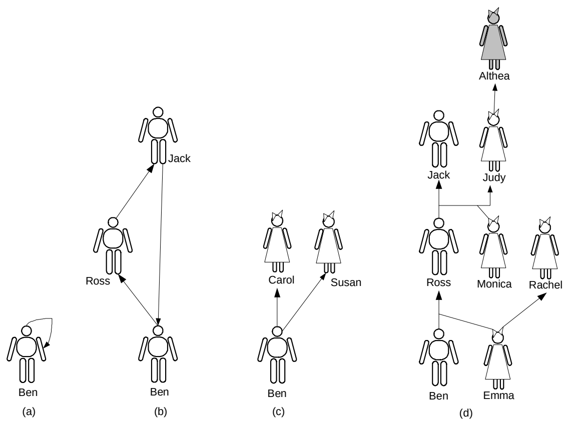


Fig 6. (a-c) Examples of invalid models and (d) of a valid model in language $\mathcal{L}_1$.

Another aspect that should be noticed is how the ontology of Figure 4 contributes for the pragmatic efficiency of $\mathcal{L}_1$. By explicitly considering the ontological meta-properties of the domain entities (e.g., if they are instantiated necessarily by their instances of not) we are able to account for other direct aspects of visual syntaxes. In the case of language $\mathcal{L}_1$, the following can be observed.

Firstly, the types Man and Woman are *kinds*. This means that instances of these types will continue to be so as long as they exist in the model. In contrast, an individual man (or woman) can have the (intrinsic) properties of *begin alive* or *being dead* in different situations. In language $\mathcal{L}_1$, the icons used to represent instances of Person maintain the stable visual percept, which represents the dichotomy of the rigid types Man and Woman. The phases living and deceased are represented as variations of this visual percept, that is, the *same* visual percept can appear in different situations as one of these two variations

Secondly, the types modeled as concrete *Roles* in the ontology are Father, Mother and Offspring. *Roles*, like phases, are instantiated by their instances only contingently. Moreover, roles are *relationally dependent* types, i.e., individuals play roles only in a certain context or in relation to another entity (Guizzardi et al., 2004). For instance, the *same* instance *x* of Father can exist in another situation in the model without being a Father. Moreover, to be a Father is to be a Man who has (at least) one Offspring, i.e., for *x* to be a Father he must share a relational property with another individual who is an instance of Offspring. In $\mathcal{L}_1$, the Parent role is represented by the adjacency relation between the icon representing a Person and the arrow-head of the symbol representing the parentOf relation. Additionally, the Offspring role is represented by the adjacency relation between the icon representing a Person and the arrow-tail of the symbol representing the parentOf relation. This representation choice highlights the dependency of these roles on relational properties. Moreover, it allows the modeling that x *qua* Man maintains its identity in the scope of different relations and across different situations.

Thirdly, the *ancestorOf* relation is represented by the above relation in the plane associated with the arrow-path-connectedness, i.e., if x and y are two persons who are path-connect and x is above y in the plane then x is an ancestorOf y. The composed relation *above-path-connect* is also irreflexive, asymmetric and transitive, i.e., a partial order relation. These are exactly the same meta-properties enjoyed by the ancestor relation. For this reason, the conclusion that (x ancestorOf z) if (x ancestorOf y) and (y ancestorOf z) is directly inferred from (x is above-path-connected-to z) if (x above-path-connected-to y) and (y above-path-connected-to z).

Final Considerations

In this paper we argue that formal domain ontologies are suitable as starting point for the design of domain-specific visual languages. We present the different elements of languages design and discuss the relation between ontologies and each of these elements. Additionally, we motivate the need for an ontologically well-justified representation language for the purpose of ontology specification and language conceptual metamodeling.

A number of formal ontologies have been developed during the years, in several important application areas. Motivated by the Semantic Web, we have seen a fast growth on the number of implemented ontologies as well as a diversification of their application domains. We demonstrate that ontology engineering can make important contributions to the area of domain-specific visual modeling languages, and that a suitably represented domain ontology can be used to: (i) provide real-world semantics for a domain language modeling primitives; (ii) delimit the set of models that should be deemed grammatically valid by that language, thus constraining its abstract syntax; and (iii) inform important pragmatic properties that should be reinforced by the language system of concrete visual syntax.

Acknowledgements. This work is part of the Freeband A-MUSE Project (contract BSIK 03025). We would like to thank Gerd Wagner, Nicola Guarino and Chris Vissers for fruitful discussions and for providing valuable input to the issues of this article.

References

Tolvanen, J.-P., Gray, J., Rossi, M., editors (2004): Domain-Specific Modeling with Visual Languages, Journal of Visual Languages and Computing, Elsevier Science.

Morris, C.W. (1938): Foundations of a theory of signs, Chicago University Press, Chicago, pp. 77-138.

Gurr, C.A. (1999): Effective Diagrammatic Communication: Syntatic, Semantic and Pragmatic Issues, Journal of Visual Languages and Computing, 10, 317-342.

Guizzardi, G.; Ferreira Pires, L; van Sinderen, M. (2005): An Ontology-Based Approach for Evaluating Domain Appropriateness and Comprehensibility Appropriateness of Modeling Languages, Proceedings of the 8th MoDELS, Jamaica.

Grice, H.P. (1975): Logic and conversation. In: Syntax and Semantics: Vol 3, Speech Acts (P. Cole & J. Morgan, eds). Academic Press, New York, pp. 43-58.

Guizzardi, G.; Wagner, G.; Guarino, N.; van Sinderen, M. (2004): An Ontologically Well-Founded Profile for UML Conceptual Models, Proceedings of the 16th CAiSE, Latvia.