Modelling as Selection of Interpretation

H.A. (Erik) Proper and Th.P. van der Weide
Institute for Computing and Information Sciences, Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, EU
{E.Proper,Th.P.vanderWeide}@cs.ru.nl

Abstract: This paper is part of an ongoing research effort to better understand the act of modelling. We describe a formal framework by which the process of modelling can be regarded as involving the selection of more and more refined interpretations in terms of the underlying meta-model of the modelling language used. The resulting framework will be used to create a laboratory setup in which we can consequently more closely study (and support) modelling processes.

1 Introduction

The work reported in this paper is part of an ongoing effort to better understand the act of modelling [HPW05a, HPW05e, PVH05, PHV05] in the context of information system engineering. One of our longer term goals is to turn the *art* of modelling into a *science* of modelling.

In the past our focus was mainly on the formal definition of syntax and semantics of modelling languages. We have recently expanded this focus to include the process of modelling and the usage of models in information systems engineering. This expansion was inspired by a desire to better understand the modelling process itself, as well as the requirements on the languages used to express these models by the context in which they are to be used [PVH05].

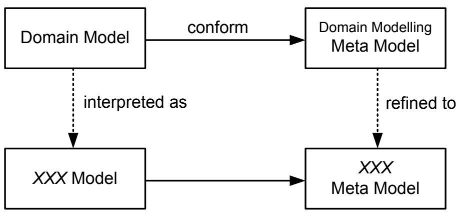

Figure 1: Refinement of models and meta-models

The primary concern of this paper is the further elaboration of a hypothesis put forward in [PHW05]. We argue that one can observe how many modelling techniques are in use to model several aspects of domains, such as processes, objects, information being processed, the flow of information, the flow of control, etc. Scholars and practitioners have produced numerous modelling techniques [Avi95, BMS98]. The resulting plethora of techniques has, in the past, already been referred to as “a methodology jungle” [Avi95]. Each of these modelling techniques focuses on specific aspects of a domain, and is especially geared towards the representation, study, analysis or design of such aspects. Nevertheless, all of these techniques deal with *facts* about a domain describing how (from the perspective of a specific aspect) concepts in the domain relate to each other. Put more operationally, we argue that any activity model, sequence diagram, information model, etc. has an accompanying domain model of the underlying concepts and their relations. Such a domain model could be expressed in terms of a general purpose domain modelling language such as ORM [Hal01], but also using ontology modelling languages such as OWL [MH03].

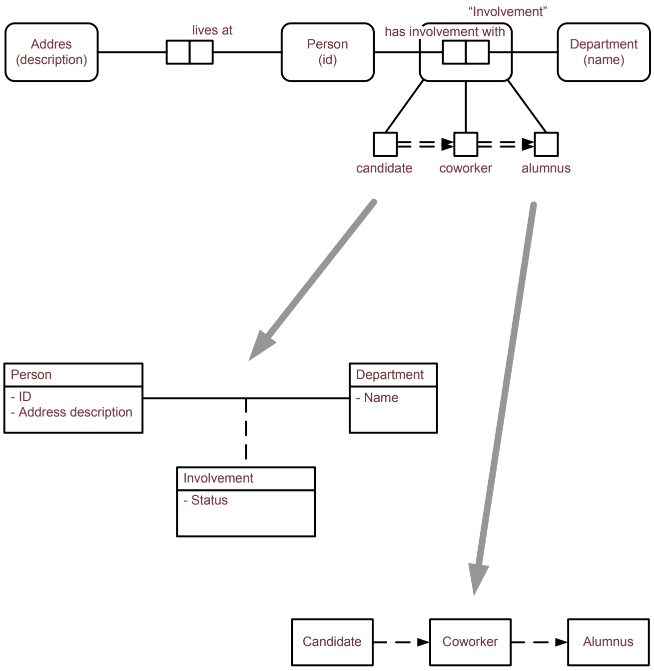

Figure 2: Example interpretations

This leads to the situation as depicted in Figure 1. On the right hand side we find the meta-models of the modelling techniques used, while on the left hand side we find the actual models. The ‘XXX’ represents an aspect of the domain that is being modelled. The ‘XXX’ model is a re-interpretation of the original model in terms of the refined ‘XXX’ meta-model. To illustrate this point, consider the example depicted in Figure 2. In this example, we have used the ORM domain modelling technique [Hal01] to represent a general domain model of a small sample domain dealing with involvement of people with a University department. The involvement starts with candidature, then might move on to the coworkership level, and will typically end in the alumnus status. In the example, we have (partially) re-interpreted the underlying domain model into two directions: an UML class diagram focussing on the core concepts in the domain, and a state-transition diagram focusing on the state changes of the involvement of people with departments.

As another example, consider the compacted version, as depicted in Figure 3, of the case study used in [PHW05]. This example focusses on workflow modelling and shows two interpretation steps. The first step, moving from A) to B), requires modelers to select which object types are really actor and actand types. The second interpretation step, from B) to C), can actually be done automatically given a pre-defined mapping between the meta-models of the modelling techniques involved. The modeller does not need to add additional information to the model. Note that the situation depicted in A) is not a *static* view on the domain. The arrows from fills in form to examines, etc, show a temporal dependency between states, thus providing a *flow* of states and activities.

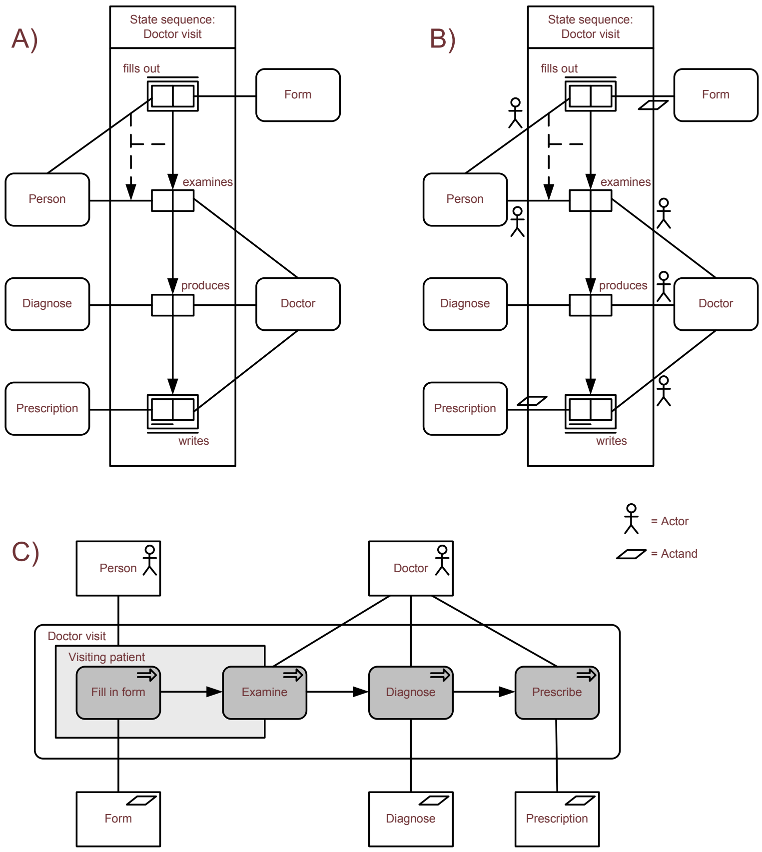

Figure 3: Activity modelling

In each of the interpretation steps, modellers need to make a choice of how to re-interpret (if at all!) specific concepts in the general domain model in terms of the modelling concepts in the refined meta-model. We argue that modelling can be regarded as a process of (iteratively!) refining ones view on the world in terms of more and more refined modelling concepts (the types in the meta-model). This process is driven by the motivations for producing the model in the first place.

One may argue that in practice, modellers will quite often directly produce UML class diagrams, workflow diagrams, etc. In our view, doing so leaves implicit numerous interpretive decisions about the domain. If one were to first produce a domain model as depicted in Figure 3 A), one could argue that the understanding of the domain being modelled would be deeper, providing a better base from which to then produce model C) via B). Note that it is not our goal not to cast judgement on how to best model. Our goal is rather to better understand the actual *act* of modelling, and as such, we do want to study how modelers implicitly or explicitly move from A) to C).

We have structured the remainder of this paper as follows. In Section 2 we briefly explore the notion of subjectivity in relation to modelling. Section Section 3 then focuses on hierarchies of modelling languages, i.e. meta-model hierarchies. Given such a hierarchy, Section 4 shows how hierarchies of models as depicted in Figure 3 can be represented formally.

2 Subjectivity in Modelling

The aim of this section is to define more precisely what we mean by the modelling of a domain, in other words, our fundamental way of thinking about modelling. In doing so, we will start by introducing a framework describing the essential processes that take place when an observer observes a domain.

It is our assumption, based on the work of C.S. Peirce [Pei69], that observers perceive a universe and then produce a *conception* of that part they deem relevant. The conceptions harboured by an observer are impossible to communicate and discuss with other observers unless they are articulated somehow (the need for this ability in the context of information systems engineering is evident). In other words, a *conception* needs to be *represented*. Peirce argues that both the perception and conception of an observer are strongly influenced by their interest in the observed universe. This leads to the following set of definitions (also inspired by the ones provided in [FVV$^+$98], which are based on the work by Peirce as well):

Universe – the 'world' around the observer.

Observer – an actor perceiving and conceiving the universe, using their senses.

Perception – that what results, in the mind of an observer, when they observe the universe, using their senses.

Conception – that what results, in the mind of a observer, when they interpret a perception of the universe.

Observers may zoom in on a particular part of the universe they observe, or to state it more precisely, they may zoom in on a particular part of their conception of the universe:

Domain of interest – any 'part' or 'aspect' of a conception of the universe, a observer may zoom in on.

In the context of information systems engineering, observers may have different domains of interest depending on their concern with regards to the information system being engineered. For example, the operators who will be required to maintain a planned information system, will regard this system in terms of costs of keeping the system up and running, costs and efforts involved in implementing the system, etc. Future users of the same planned system, however, will be more interested in the impact/support the system is likely to have on their work related tasks. In our effort to obtain a fundamental understanding of the act of modelling, we initially focus on situations where we only have one specific concern and associated domain of interest. In line with [FVV$^+$98] we define a model to be a specific kind of conception:

Model – a purposely abstracted and unambiguous conception of a domain of interest.

Conceptions that are harboured by an observer are impossible to communicate and discuss with other observers, unless they are articulated somehow. In other words, the conception needs to be *represented*:

Representation – the result of an observer representing a conception, using some language to express themselves.

The resulting situation is illustrated in Figure 5 showing how an observer in observing the universe has a conception, which may be represented in terms of a representation. We are now also in a position to define more precisely what we mean by modelling:

Modelling – The act of purposely forming a model from (what is conceived to be) a part of the universe, and *representing* the resulting model by means of some language and medium.

In the context of information systems engineering, observers will approach a domain with the aim of expressing the domain in terms of some set of modelling constructs, such as classes, activity (types), event (types), constraints, etc. The set of modelling constructs a observer is used to employ (or trained to use) when modelling a domain, will strongly influence his/her conceptions. For example, when viewing a domain of interest from the perspective of UML class diagrams, this is bound to lead to a different model than when the same domain is viewed from the perspective of UML sequence diagrams. To make this explicit, we therefore presume that when observers model a domain, they do so from a certain perspective; their *Weltanschauung* [WAA85]. Figure 5 also illustrates how an observer observes (a domain of interest within) a universe from the perspective of different meta-models ($M_1, \ldots, M_n$), leading to equally many models ($m_1, \ldots, m_n$) and model representations ($r_1, \ldots, r_n$).

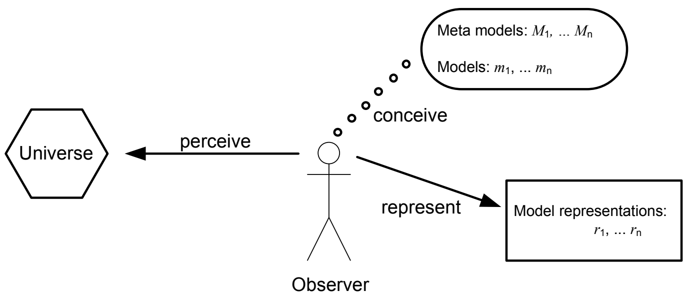

Figure 4: Observing a universe with different meta-models

The remainder of this paper is primarily concerned with the development of a precise understanding of the relationships between these meta-models, the corresponding models (or rather their representations), as well as their evolution during a modelling process. Here we will operate under the hypothesis that modelling can be viewed as an iterative process of: (1) defining an (unspecific) model of a domain using some suitable generic meta-model, focussing on domain concepts and their relationships in a general sense, (2) selecting more specific interpretations of the concepts identified in the initial model using more refined meta-models. The latter step, selection of interpretation, is the essential aspect of the way of thinking as put forward in this paper, and is therefore also used as the title for this paper.

3 Meta-model Hierarchies

The foundation of our modelling framework is formed by a hierarchy of meta-models. The concept of a *meta-model hierarchy* is not new. It was already introduced in [FO94] as a way of comparing modelling techniques, and to some extent refined further in [FVV$^+$98]. Our goals of viewing the act of modelling as a process of stepwise selection of interpretations over a hierarchy of meta-models is a way to operationalise the 'old' notion of a meta-model hierarchy.

A meta-model is seen as a formal system [Men87]. Such a system consists of (1) a signature that specifies its concepts, providing a base for the definition of well-formed formulae, and (2) a set of such well-formed formulae (also refered to as axioms) that are assumed/required to hold for concrete systems that realize the formal system. In this context we shall refer to the concepts of the formal system as the (modeling) types of the meta-model. We will denote a meta-model by its signature and its axioms. We will use $\langle T, A\rangle$ to denote the system with signature T and axioms A.

Let $\mathcal{MT}$ be the set of all meta-types from some class of modelling techniques, $\mathcal{MA}$ be the set of all axioms, and $\mathcal{MM} \subseteq \mathcal{MT} \times \mathcal{MA}$ the set of all meta-models. We focus on meta-models that satisfy the following rules. Each meta-model is consistent, meaning that the axioms are not contradictory.
[M1] If $\langle T, A\rangle \in \mathcal{MM}$, then A is a consistent set of well-formed formula's over T.
Each meta-model is required to have different modeling types.
[M2] If $M_1 = \langle T_1, A_1\rangle$ and $M_2 = \langle T_2, A_2\rangle$, such that $M_1, M_2 \in \mathcal{MM}$, then:
$M_1 \neq M_2 \Rightarrow T_1 \cap T_2 = \emptyset$
This latter requirement is added to allow us to study relations between modeling concepts in more depth.

A model is regarded as an instantiation of a formal system; the associated meta-model. This model thus contains instantiations of the meta-types contained in that meta-model. Let $\mathcal{EL}$ be the set of all those instantiations, which are referred to as model elements. We define the possible interpretation of these elements in terms of the meta-types: $\mathcal{IN} = \mathcal{EL} \times \mathcal{MT}$. In other words, an interpretation is the combination of a model element and a meta-type. Since meta-models may contain sub-types, elements may be associated to multiple meta-types.

If m is a model with associated meta-model M, we will also say that m is an M-model. An M-model m can be regarded as a set of interpretations $m \subseteq \mathcal{IN}$ that meet the axioms of meta-model M. The set of valid M-models for a given meta-model $M = \langle T, A \rangle$ is therefore defined as: $\mathbb{M}(M) \triangleq \{m \subseteq \mathcal{EL} \times T \mid m \models A\}$. The set of interpretations fitting a meta-model is defined as: $\mathbb{I}(M) \triangleq \cup \mathbb{M}(M)$.

The next step is to introduce hierarchies of meta-models. Such a hierarchy is composed of refinement relations between meta-models. Let $\mathcal{RF}$ be the set of possible refinement relations for the considered class of meta-models and let $\mathsf{From}, \mathsf{To} : \mathcal{RF} \rightarrow \mathcal{MM}$ be functions returning the start and destination meta-model of a refinement respectivily. Then $\mathcal{RF}$, From and To together span a space in which we will be able to identify meta-model hierarchies to be used in modelling. We do require $\mathcal{RF}$ to be acyclic:
[M3] The graph spanned over $\mathcal{MM}$ by From and To is acyclic.
A specific meta-model hierarchy is a set of refinements, so we can define the set of possible meta-model hierarchies as $\mathcal{MH} \subseteq \wp(\mathcal{RF})$, where we do require:
[M4] If $R \in \mathcal{MH}$ then R is a tree.
Let $\mathsf{Top}(R)$ denote the top of such a tree. We will write $R_{\mathcal{MM}}$ as an abbreviation for the set of meta-models involved in R.

To really capture the notion of refinement between meta-models, we must be able to map models upward in the hierarchy. We therefore need a function that is able to ground models stated in a refined meta-model in terms of the more general meta-model:

$$\mathsf{Ground}_{\_} : \mathcal{RF} \rightarrow (\wp(\mathcal{IN}) \rightarrow \wp(\mathcal{IN}))$$

In terms of the example shown in Figure 3 the grounding function would have to map any actor type and actand type in a workflow model onto an object type in an ORM model, and each activity type onto an ORM relationship type. The working of the grounding function is illustrated in Figure 6. Models are grounded by grounding the interpretations they are made of. Multiple models conform a refined meta-model may be grounded onto the same generalized model. For example, in Figure 3 we might have selected a person being examined to be an actand (i.e. passive) in the examination, rather than considering it to be an actor as well (as is currently shown in B)). In either case, the grounding of model B) would still be the model shown in A).

For a given refinement r, the grounding function should limit itself to interpretations associated to the meta-models involved in the refinement:
[M5] $x \in \mathtt{dom}(\mathsf{Ground}_r) \Rightarrow x \subseteq \mathbb{I}(\mathsf{To}(r))$ and $y \in \mathtt{ran}(\mathsf{Ground}_r) \Rightarrow y \subseteq \mathbb{I}(\mathsf{From}(r))$
Empty models have an empty grounding:
[M6] $\mathsf{Ground}_r(\emptyset) = \emptyset$
Even more, the grounding function should behave strict monotonous in terms of inclusion of sets of interpretations:
[M7] $m_1 \subset m_2 \subseteq \mathcal{IN} \Rightarrow \mathsf{Ground}_r(m_1) \subset \mathsf{Ground}_r(m_2)$
where $\subset$ is used as a *proper* subset. This allows us to ground any non-empty fragment of a re-interpreted model back to (a non-empty) fragment at the more generic level:
Corollary 3.1 $m \neq \emptyset \Rightarrow \mathsf{Ground}_r(m) \neq \emptyset$

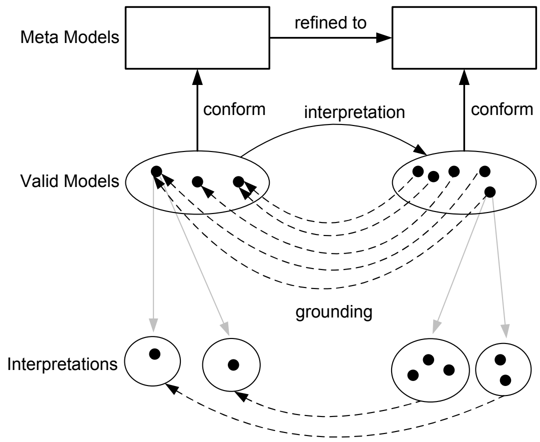

Figure 5: Grounding of models and interpretations

4 Model Hierarchies

In this section we extend the meta-model hierarchy of the previous section to a hierarchy of models over such meta-model hierarchies. First we follow the interpretation of a single model element in a hierarchy. When modelling, decisions are made pertaining to interpretations of the domain. These modeling decisions are almost as important as the resulting model. Let $\mathcal{MV}$ be a carrier set for motivations of such decisions, then we can define an interpretation hierarchy as a *partial* function: $h : \mathcal{MM} \rightarrowtail \wp^+(\mathcal{IN}) \times \mathcal{MV}$. Let $\mathcal{IH}$ be the set of all such interpretation hierarchies. If we are only interested in the set of interpretations, we will use: $h!(M) \triangleq I$ **such that** $h(M) = \langle I, v \rangle$.

An interpretation hierarchy should follow a meta-model hierarchy. This is laid down in three rules. We consider h to be an interpretation hierarchy fitting a meta-model hierarchy R, written as $h \in I(R)$, iff:
(1) An interpretation hierarchy can only contain interpretations for the meta-models present in R. Formally: $\mathtt{dom}(h) \subseteq R_{\mathcal{MM}}$.
(2) The top of the interpretation hierarchy is required to contain one interpretation only; the root. Formally: $|h!(\mathsf{Top}(R))| = 1$.
(3) The interpretation hierarchy is required to obey the grounding function. Formally this is enforced by: $\forall_{r \in R} [\mathsf{Ground}_r(h!(\mathsf{To}(r))) \subseteq h!(\mathsf{From}(r))]$.

A model hierarchy is a set H of interpretation hierarchies. If H is a model hierarchy, then for any meta-model M, the complete model is defined as the union of the interpretations in the interpretation hierarchies (as illustrated in Figure 7):

$$H!(M) \triangleq \bigcup_{h \in H} h!(M)$$

For a given meta-model hierarchy R, the set of valid model hierarchies consists of those interpretation hierarchies H such that:

$$\forall_{M \in \mathtt{dom}(H!)} [H!(M) \in \mathbb{M}(M)] \wedge \forall_{h,i \in H} [h \neq i \Rightarrow h \otimes i]$$

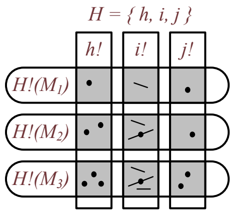

Figure 6: Models as a union of interpretations

The first condition requires that all models in the hierarchy conform to their respective meta-models, while the second condition requires interpretation hierarchies to not overlap. Two interpretation hierarchies are disjoint iff they do not overlap for any meta-model:
$h \otimes i \triangleq \forall_{M \in \mathcal{MM}} [h!(M) \cap i!(M) = \emptyset]$

5 Evolution of Model Hierarchies

During a modelling process, a model hierarchy is likely to evolve. The final step in the definition of our framework is therefore the introduction of a temporal dimension. This essentially boils down to the extension of the framework so-far with a schema evolution mechanism as defined before in e.g. [PW94, Pro97]. Within the confines of this paper, we can only provide a brief overview of how evolution of model hierarchies can be formalized.

Let $\mathcal{TI}$ be a set of points in time with a total ordering $<$. Let furthermore $\rhd$ be a function returning the next point in time, such that:
[T1] $\forall_{t \in \mathcal{TI}} [t < \rhd t]$

[T2] $\neg \exists_s [t < s < \rhd t]$
The evolution of a model hierarchy is modelled by tracing the evolution of the underlying interpretation hierarchies. The latter evolutions are modelled as a function $e : \mathcal{TI} \to \mathcal{IH}$. Let $\mathcal{IE} = \mathcal{TI} \to \mathcal{IH}$ be the set of possible evolutions of interpretation hierarchies. The evolution of a model hierarchy can then be regarded as a subset $E \subseteq \mathcal{IE}$. At any point in time, the current version of the model hierarchy can be derived as:

$$E(t) \triangleq e(t)$$

The set of valid model hierarchy evolutions for a given meta-model hierarchy R is defined as:

$$\mathbb{E}(R) \triangleq \left\{ E \subseteq \mathcal{IE} \;\middle|\; \begin{array}{ll} \forall_{t \in \mathcal{TI}} [E(t) \in \mathbb{M}(R)] & \wedge \\ \forall_{e \in E} [e(t) \subseteq e(\rhd t) \vee e(t) \supseteq e(\rhd t)] & \end{array} \right\}$$

The first condition requires evolutions to obey the meta-model hierarchy. The second condition is used to ensure that the steps in the modelling process are not too large, retracting and extending a hierarchy cannot be done in one step.

In line with [Pro97], information providing motivation for may be added to each of the evolution steps. To be able to do so, we enrich the hierarchy evolutions with a partial function:

$$\mathsf{Motivation} : \wp(\mathcal{IE}) \times \mathcal{TI} \times \wp(\mathcal{IE}) \rightarrowtail \mathcal{MV}$$

where the intuition is that if $\mathsf{Motivation}(E, t, C) = o$, then a change occuring within a hierarchy evolution E at time t, involving the interpretation evolutions C is motivated by o.

We require the movitations for changes are limited to those interpretation evolutions that are part of the given hierarchy evolution:
[M8] $\langle E, t, C\rangle \in \mathtt{dom}(\mathsf{Motivation}) \Rightarrow C \subseteq E$
The reason for C to be a set of evolutions has to do with the fact that during the modelling process a modeler may decide to merge and/or split existing evolutions. In [Pro97] a terminology based on particle physics was introduced to identify such situations as fusion, fission, emission and absorption.

We require all changes to be motivated:
[M9] If $E \in \mathbb{E}(R)$ then:

$$\forall_{e \in E} \left[e(t) \neq e(\rhd t) \Rightarrow \exists_C \left[e \in C \wedge \langle E, \rhd t, C\rangle \in \mathtt{dom}(\mathsf{Motivation})\right]\right]$$

Only changes can be motivated:
[M10] $\langle E, t, C\rangle \in \mathtt{dom}(\mathsf{Motivation}) \Rightarrow \forall_{e \in C} \left[e(t) \neq e(\rhd t)\right]$
Consider this:

> Take a closer look at this. In [Pro97] we allowed for interpretation evolutions to participate as catalysts. In this case we would require $\exists_{e \in C} \left[e(t) \neq e(\rhd t)\right]$.

Consider this:

6 Enabling the Modelling Process

Lijn: Logbook is dialoog georienteerde perspectief op modelleren. Modellerproces is een Q/A proces. Deel van dit proces wordt gedreven door het doel om een model hierarchy te populeren. Daar kunnen we onderstaande middelen voor inzetten.

Sometimes we have automatic mappings. They may use heuristics in doing an auto interpretation. Note that the result must still be groundable! A mapping using ONF of a conceptual schema is *not* groundable unless the ONF also contains the original conceptual model.

Dit moet nog geschreven worden.

Idee is: Laat een op logica regels gebaseerde grounding zien. Dat geeft aan dat je er mee zou kunnen redeneren. Bijvoorbeeld:

```
ActivityType(x) => FactType(x)

ActorType(x) => ObjectType(x)

ActandType(x) => ObjectType(x)

InvolvementType(r) => RoleType(r) AND Name(r,"involved in")

InvolvementIn(r,x) => HasPlayer(r,x)

InvolvementWith(r,x) => HasFact(r,x)
```

Verder kunnen we aangeven dat een modelleeur bij het seelcteren typisch als keuze zal hebben voor een gegeven model(fragment): $\mathsf{Choice}_r(m) \triangleq \left\{n \,\middle|\, \mathsf{Ground}_r(n) = m\right\}$

Note: Add text about the modelling goals as well. A hierarchy may be partially complete. In the modelling process, one could drive the interaction based on the knowledge the system has about the meta-models.

Logbook link!!

Consequence:

Modeling process can be done linearly: First produce an ORM model Then specialize this to an XXX model

But also inversely Create XXX model Implies underlying ORM model

And iteratively Switch between elaboration of XXX and ORM model

Utilizing the specialization relationship between an ORM model and a more specific XXX model would: Allow us to use ORM rigor in identifying the concepts when creating an XXX model Re-use ORMs natural language foundations when creating an XXX model Confront modelers more fundamentally with decisions about concepts and relations

7 Conclusion

In this paper we have discussed a framework to study the act of modelling, where a modelling process is regarded as involving the selection of more and more refined interpretations in terms of the underlying meta-model of the modelling language used. The resulting framework will be used, in conjunction with the logbook system, to create a laboratory environment in which modelling experiments can be conducted.

The logbook system [HPW05a] takes the view that a modelling process is a (controlled) dialogue between a domain expert, a modelling mediator and a model builder. This process is regarded as a questioning & answering process involving these three roles. When combined with the theory as presented in this paper, the goal of such a questioning & answering process can be made explicit as the creation of a model hierarchy on top of a pre-determined (dictated by the modelling goals at hand [PVH05, PHV05]) meta-model hierarchy.

In future versions of our framework we also intend to refine it such that we are able to deal with multiple views and concerns, as well as multiple (contradicting!) observers. In the latter case we would like to be able to even log the negotiation that may have to take place in reconciling different views held by different observers of the same domain.

References

[Avi95] D.E. Avison. *Information Systems Development: Methodologies, Techniques and Tools*. McGraw–Hill, New York, New York, USA, 2nd edition, 1995. ISBN 0077092333.

[BBMP95] G.H.W.M. Bronts, S.J. Brouwer, C.L.J. Martens, and H.A. (Erik) Proper. A Unifying Object Role Modelling Approach. *Information Systems*, 20(3):213–235, 1995.

[BFW96] P. van Bommel, P.J.M. Frederiks, and Th.P. van der Weide. Object–Oriented Modeling based on Logbooks. *The Computer Journal*, 39(9):793–799, 1996.

[BHW91] P. van Bommel, A.H.M. ter Hofstede, and Th.P. van der Weide. Semantics and verification of object–role models. *Information Systems*, 16(5):471–495, October 1991.

[BMS98] P. Bernus, K. Mertins, and G. Schmidt, editors. *Handbook on Architectures of Information Systems*. International Handbooks on Information Systems. Springer, Berlin, Germany, EU, 1998. ISBN 3540644539.

[BPH04] A.I. Bleeker, H.A. (Erik) Proper, and S.J.B.A. Hoppenbrouwers. The Role of Concept Management in System Development – A practical and a theoretical perspective. In J. Grabis, A. Persson, and J. Stirna, editors, *Forum proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia, EU*, pages 73–82, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology. ISBN 998497670X.

[Bub86] J.A. Bubenko. Information System Methodologies – A Research View. In T.W. Olle, H.G. Sol, and A.A. Verrijn–Stuart, editors, *Information Systems Design Methodologies: Improving the Practice, Amsterdam, The Netherlands, EU*, pages 289–318. North–Holland/IFIP WG8.1, Amsterdam, The Netherlands, EU, 1986.

[BW04a] S. Bosman and Th.P. van der Weide. Assistance for the domain modeling dialog. Technical Report NIII–R0421, Computing Science Institute, University of Nijmegen, Nijmegen, The Netherlands, EU, 2004.

[BW04b] S. Bosman and Th.P. van der Weide. Towards formalization of the information modeling dialog. Technical Report ICIS–R05012, Computing Science Institute, University of Nijmegen, Nijmegen, The Netherlands, EU, 2004.

[CHP96] L.J. Campbell, T.A. Halpin, and H.A. (Erik) Proper. Conceptual Schemas with Abstractions – Making flat conceptual schemas more comprehensible. *Data & Knowledge Engineering*, 20(1):39–85, 1996.

[CP96] P.N. Creasy and H.A. (Erik) Proper. A Generic Model for 3–Dimensional Conceptual Modelling. *Data & Knowledge Engineering*, 20(2):119–162, 1996.

[DFW96] C.F. Derksen, P.J.M. Frederiks, and Th.P. van der Weide. Paraphrasing as a Technique to Support Object–Oriented Analysis. In R.P. van der Riet, J.F.M. Burg, and A.J. van der Vos, editors, *Proceedings of the Second Workshop on Applications of Natural Language to Databases (NLDB'96)*, pages 28–39, June 1996.

[FO94] E.D. Falkenberg and J.L.H. Oei. Meta Model Hierarchies from an Object–Role Modelling Perspective. In T.A. Halpin and R. Meersman, editors, *Proceedings of the First International Conference on Object–Role Modelling (ORM–1)*, pages 218–227, July 1994.

[FVV+98] E.D. Falkenberg, A.A. Verrijn–Stuart, K. Voss, W. Hesse, P. Lindgreen, B.E. Nilsson, J.L.H. Oei, C. Rolland, and R.K. and Stamper, editors. *A Framework of Information Systems Concepts*. IFIP WG 8.1 Task Group FRISCO, IFIP, Laxenburg, Austria, EU, 1998. ISBN 3901882014.

[FW04] P.J.M. Frederiks and Th.P. van der Weide. Information Modeling: the process and the required competencies of its participants. In F. Meziane and E. Métais, editors, *9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004), Manchester, United Kingdom, EU*, volume 3136 of *Lecture Notes in Computer Science*, pages 123–134, Berlin, Germany, EU, 2004. Springer.

[Hal01] T.A. Halpin. *Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design*. Morgan Kaufmann, San Mateo, California, USA, 2001. ISBN 1558606726.

[HBP05] S.J.B.A. Hoppenbrouwers, A.I. Bleeker, and H.A. (Erik) Proper. Facing the Conceptual Complexities in Business Domain Modeling. *Computing Letters*, 1(2):59–68, 2005.

[HP04] S.J.B.A. Hoppenbrouwers and H.A. (Erik) Proper. A Communicative Perspective on Second Order Information Systems. In G.E. Lasker, editor, *Proceedings of the 16th International Conference on System Research, Informatics and Cybernetics, Baden–Baden, Germany*. IIAS, 2004.

[HPR05] S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and V.E. van Reijswoud. Navigating the Methodology Jungle – The communicative role of modelling techniques in information system development. *Computing Letters*, 1(3), 2005.

[HPW93] A.H.M. ter Hofstede, H.A. (Erik) Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.

[HPW05a] S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. A Fundamental View on the Process of Conceptual Modeling. In L. Delcambre, C. Kop, H.C. Mayr, J. M ylopoulos, and O. Pastor, editors, *Conceptual Modeling – ER 2005 – 24 International Conference on Conceptual Modeling, Klagenfurt, Austria, EU*, volume 3716 of *Lecture Notes in Computer Science*, pages 128–143, Berlin, Germany, June 2005. Springer–Verlag. ISBN 3540293892 `doi:10.1007/11568322_9`.

[HPW05b] S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. Fact Calculus: Using ORM and Lisa–D to Reason About Domains. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2005: OTM Workshops – OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005, Agia Napa, Cyprus, EU*, volume 3762 of *Lecture Notes in Computer Science*, pages 720–729, Berlin, Germany, October/November 2005. Springer–Verlag. ISBN 3540297391 `doi:10.1007/11575863_91`.

[HPW05c] S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. Formal Modelling as a Grounded Conversation. In G. Goldkuhl, M. Lind, and S. Haraldson, editors, *Proceedings of the 10th International Working Conference on t he Language Action Perspective on Communication Modelling (LAP'05)*, pages 139–155, Kiruna, Sweden, EU, June 2005. Linköpings Universitet and Hogskolan I Boras, Linköping, Sweden, EU.

[HPW05d] S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. Towards explicit strategies for modeling. In T.A. Halpin, K. Siau, and J. Krogstie, editors, *Proceedings of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'05), held in conjunctiun with the 17th Conference on Advanced Information Systems 2005 (CAiSE 2005)*, pages 485–492, Porto, Portugal, EU, 2005. FEUP, Porto, Portugal, EU. ISBN 9727520774.

[HPW05e] S.J.B.A. Hoppenbrouwers, H.A. (Erik) Proper, and Th.P. van der Weide. Understanding the Requirements on Modelling Techniques. In O. Pastor and J. Falcao e Cunha, editors, *17th International Conference on Advanced Information Systems Engineering, CAiSE 2005, Porto, Portugal, EU*, volume 3520 of *Lecture Notes in Computer Science*, pages 262–276, Berlin, Germany, June 2005. Springer–Verlag. ISBN 3540260951 `doi:10.1007/11431855_19`.

[HVH97] J.J.A.C. Hoppenbrouwers, B. van der Vos, and S.J.B.A. Hoppenbrouwers. NL Structures and Conceptual Modelling: Grammalizing for KISS. *Data & Knowledge Engineering*, 23(1):79–92, 1997.

[HW93] A.H.M. ter Hofstede and Th.P. van der Weide. Expressiveness in conceptual data modelling. *Data & Knowledge Engineering*, 10(1):65–100, February 1993.

[Men87] E. Mendelson. *Introduction to Mathematical Logic*. Wadsworth and Brooks, 1987.

[MH03] D.L. McGuiness and Frank van Harmelen. *OWL Web Ontology Language Overview, W3C Proposed Recommendation*. W3C, December 2003. `http://www.w3.org/TR/2003/PR-owl-features-20031215/`.

[Myl98] J. Mylopoulos. Techniques and Languages for the Description of Information Systems. In P. Bernus, K. Mertins, and G. Schmidt, editors, *Handbook on Architectures of Information Systems, Berlin, Germany, EU*. Springer, Berlin, Germany, EU, international handbooks on information systems edition, 1998. ISBN 3540644539.

[PBH04] H.A. (Erik) Proper, A.I. Bleeker, and S.J.B.A. Hoppenbrouwers. Object–Role Modelling as a Domain Modelling Approach. In J. Grundspenkis and M. Kirikova, editors, *Proceedings of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'04), held in conjunctiun with the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004),*, volume 3, pages 317–328, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology. ISBN 9984976718.

[Pei69] C.S. Peirce. *Volumes I and II – Principles of Philosophy and Elements of Logic*. Collected Papers of C.S. Peirce. Harvard University Press, Boston, Massachusetts, USA, 1969. ISBN 0674138007.

[PH04] H.A. (Erik) Proper and S.J.B.A. Hoppenbrouwers. Concept Evolution in Information System Evolution. In J. Gravis, A. Persson, and J. Stirna, editors, *Forum proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia, EU, Riga, Latvia, EU*, pages 63–72, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology. ISBN 998497670X.

[PHV05] H.A. (Erik) Proper, S.J.B.A. Hoppenbrouwers, and G.E. Veldhuijzen van Zanten. Communication of Enterprise Architectures. In M.M. Lankhorst, editor, *Enterprise Architecture at Work: Modelling, Communication and Analysis, Berlin, Germany, EU*, pages 67–82. Springer, Berlin, Germany, EU, 2005. ISBN 3540243712.

[PHW05] H.A. (Erik) Proper, S.J.B.A. Hoppenbrouwers, and Th.P. van der Weide. A Fact–Oriented Approach to Activity Modeling. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2005: OTM Workshops – OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005, Agia Napa, Cyprus, EU*, volume 3762 of *Lecture Notes in Computer Science*, pages 666–675, Berlin, Germany, October/November 2005. Springer–Verlag. ISBN 3540297391 `doi:10.1007/11575863_86`.

[Pro97] H.A. (Erik) Proper. Data Schema Design as a Schema Evolution Process. *Data & Knowledge Engineering*, 22(2):159–189, 1997.

[PVH05] H.A. (Erik) Proper, A.A. Verrijn–Stuart, and S.J.B.A. Hoppenbrouwers. Towards Utility–based Selection of Architecture–Modelling Concepts. In S. Hartmann and M. Stumptner, editors, *Proceedings of the Second Asia–Pacific Conference on Conceptual Modelling (APCCM2005), Newcastle, New South Wales, Australia*, volume 42 of *Conferences in Research and Practice in Information Technology Series*, pages 25–36, Sydney, New South Wales, Australia, January 2005. Australian Computer Society. ISBN 1920682252.

[PW94] H.A. (Erik) Proper and Th.P. van der Weide. EVORM – A Conceptual Modelling Technique for Evolving Application Domains. *Data & Knowledge Engineering*, 12:313–359, 1994.

[VHP04] G.E. Veldhuijzen van Zanten, S.J.B.A. Hoppenbrouwers, and H.A. (Erik) Proper. System Development as a Rational Communicative Process. *Journal of Systemics, Cybernetics and Informatics*, 2(4), 2004. `http://www.iiisci.org/Journal/sci/pdfs/P492036.pdf`.

[WAA85] A.T. Wood–Harper, L. Antill, and D.E. Avison. *Information Systems Definition: The Multiview Approach*. Blackwell, Oxford, United Kingdom, EU, 1985. ISBN 0632012168.

[WS92] W. Woods and J. Schmolze. The KL–ONE family. *Computers and Mathematics with Applications*, 23(2–5):133–177, 1992.