

# Meta-Modelling as a Means for Improved Communication and Interoperability – The Case of Frisco

Petia Wohed<sup>1</sup> and Birger Andersson<sup>2</sup>

<sup>1</sup> Centre de Recherche en Automatique de Nancy  
Université Henri Poincaré, Nancy 1/CNRS  
BP239, 54506 Vandoeuvre les Nancy, France  
petia.wohed@cran.uhp-nancy.fr

<sup>2</sup> Department of Computer and Systems Sciences  
Stockholm University/The Royal Institute of Technology  
Forum 100, SE-164 40 Kista, Sweden  
ba@dsv.su.se

**Abstract.** Unclear and disparate terminology are causing problems in the area of information system development. To address these issues, modelling frameworks and ontologies based on solid philosophical and logical foundations have been developed. Hypothesizing that a diagrammatic interface will be beneficial for comprehension and communication resulting in wider dissemination, we propose a UML-based meta-model of one of those frameworks—Frisco. The analysis done during the development of the meta-model resulted in identification of a number of ambiguities in the framework. Discussion of these as well as suggestions of ways to solve them are also presented.

## 1 Introduction

Terminological fuzziness is characterizing the field of information systems creating all kinds of problems. To address this, during the last decades, frameworks like Frisco (Framework of Information Systems Concepts) [1] and BWW (Bunge, Wand and Weber models) [2] were developed. The goals of those frameworks coincides with the goals of ontology analysis and engineering [3], namely: to provide a coherent system of concepts, the use of which would reduce misunderstandings and ambiguities and would 1) support communication between stakeholder with different needs and approaches 2) facilitate interoperability among systems and 3) be beneficial for requirement identification, reusability and reliability within information system development process. Thus, the field of ontology engineering has received a growing attention where ontologies like CYC [4] in the knowledge representation area, and Toronto Virtual Enterprise (TOVE) and Enterprise Ontology (EO) [5, 6] for enterprise modelling have been developed.

Characteristic for all these approaches is that they rely on solid theoretical work, are developed by mathematicians, logicians and/or computer scientists, are well formalised and their initial representation is in some formal notation.

While this last property is an absolute requirement for achieving interoperability and clear semantics, it is hardly beneficial when it comes to the fulfilment of the communication goal mentioned above. Not every person involved in an information system development process can be assumed to be skilled in reading and understanding formal notations.

To address the issue of comprehensibility, graphical formalisms has been developed. For instance, in [7] a graphical meta-model was developed for representation of BWW models. In this paper we report on using this approach when developing such a meta-model for the Frisco framework. As in [7] we motivate the approach by the following arguments.

- The development of a meta-model that visualises the concepts currently existing and the framework and the relationships between them will increase the communication and understanding of the framework and facilitate its use.
- The development of a meta-model will support analysis of the framework and facilitate a further development of it.
- The existence of a meta-model will further facilitate analysis of Frisco with respect to other existing approaches e.g., BWW, Tove, EO, etc. During such analysis overlapping and discrepancies between the ontologies shall be outlined which will result in a clearer positioning of Frisco according to the other relevant works in the area.
- Meta-models of ontologies, in general, support the analysis of modelling languages in the sense that they can be used as reference frameworks through which the considered languages are analysed.
- The experience of the work on the development and utilisation of the meta-model will contribute to the overall knowledge within Ontology Engineering field.

Unlike [7], where the benefits of the developed meta-model were mainly demonstrated for the purposes of the analysis of two grammars, i.e., according to the fourth point outlined above, and only briefly exemplifying the second point above, we are in this paper in detail demonstrating how the development of a meta-model can benefit for the further development of a framework. We have selected Frisco to be the subject for our work because, 1) it was specifically developed for addressing the conceptualization within the Information Systems domain, 2) we are interested in an cross-analysis between FRISCO and BWW (and later on with EO and TOVE). As an initial step in such a cross-analysis, the creation and use of graphical meta-models, e.g. the one developed for BWW or the one for Frisco partially presented here, will be beneficial.

As a graphical representation language, we are using UML class diagrams [8]. This choice is motivated by the wide dispersion and knowledge of UML, often also referred to as a de-facto standard for information system analysis and development. UML class diagrams limits our expressive power to what can be stated using those diagrams, but for the purposes of this paper they are adequate.

The paper proceeds by presentation of Frisco and the development of meta-model for it in the next section. In section 3, the results of the analysis of Frisco

provided during the development of the meta-model are presented and some improvements suggested. Section 4, concludes the work and outlines directions for further research.

## 2 The Frisco framework

As any other axiomatic system, Frisco starts by defining a small set of primitives from which the rest of the concepts are gradually built up. It is divided into a number of layers: a fundamental layer, an actor, actions and actands layer, a system concepts layer, and an organizational and information system concepts layer. Due to space limitation, in this paper we only present and work on the the fundamental and actor, actions, and actands layers, which we discuss below.

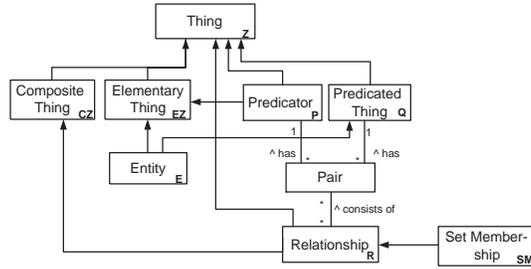
### 2.1 The Fundamental Layer

The world is made up of **Things**. A thing is either an **Elementary** or a **Composite thing**. Composite things are build up through **Relationships**. Relationships are sets of binary tuples, the elements of which are things: the first element in a tuple is called a **Predicated thing** and the second element a **Predicator**. Relationships are themselves considered as things, which makes it possible to represent the complex structures often existing in a domain.

The formal definitions are reprinted from [1] in table 1 in the Appendix. The meta-model which we propose is drawn in figure 1. Generally, every concept is represented by a class and in the lower right hand corner of every class the name of the set as defined in table 1 is given.

The relationships between concepts are captured either by generalisation/specialisation constraints (i.e., isa relationships) or by ordinary relationships to which we have given names either appearing in the formal definitions, or just close to the natural language comprehension. The cardinality constraints are derived from the formal definitions. Some rules like that “A Thing is either Elementary or Composite” are directly captured in the graphical model (by joining the ends of the corresponding isa relationships) while others like that “An Elementary Thing is defined as a Thing which is not a Relationship...” can rather indirectly be seen in the graphical model through the implications they bring e.g., “The sets of Elementary things and Relationships are disjoint” and through the reasoning that Relationships are kinds of (isa) Composite Things which in turns are disjoint with the Elementary Things. Furthermore, for the sake of explicitness we have introduced the class **Pair**, which does not have a formal definition of its own. However, the concept **Pair** is used for defining Relationships and distinguishing between unary relationships (consisting of one pair), binary (i.e., relationships composed of a couple of pairs), ternary relationships (consisting three pairs), and so on.

According to the description in Frisco, **Predicator** and **Predicated Things** do not need to be disjoint sets (so we have not joined their isa relationships endings) but in most real life cases they are indeed disjoint. Furthermore, the isa relationship between **Relationship** and **Thing** classes is derivable through the



**Fig. 1.** Frisco - basic concepts

transitivity of the isa relationships from Relationship to Composite Thing to Thing, but we have chosen to explicitly draw it in the model to directly depict the definition of Relationship concept stating that a relationships is a thing. Finally for this part of the framework, a Set Membership is a binary relationship with the predicators has-element and is-element-of within its first and second pair correspondingly. As instances of the class Predicator has-element and is-element-of are not directly depicted in the graphical model. Also the relationship between Elementary Thing, as consisting of Things that are not Relationships and that does not appear in pairs with the predicator has-element has been difficult to capture graphically and is for the moment missing from the model.

In the remainder of the Fundamental layer the concept of Transition with surrounding terminology is introduced. (The formal definitions are reprinted in table 2 in the Appendix). A Transition is a kind of binary relationship in which the predicators in the tuples are the primitives before and after and the predicated things are composite things, so called States. Complex transitions can be built up through Sequence, Choice and Concurrency to State Transitions Structures. A coherent state transition structure, i.e., a structure with a unique input (before) state and a unique output (after) state, is called a Composite Transition. Furthermore, Rules are used to define the set of permissible states and transitions in a context. To capture these concepts we are gradually extending the meta-model with the shaded classes as depicted in figure 2.

In this part of the framework the concept of Transition Occurrence for capturing the different occurrences of the transitions is introduced. We have introduced the corresponding class into the model, but we leave the discussion of it for the next section. The concepts Type, Population and Instance, which are all part of the formal definition of the framework, are for the moment left out of the model but will be discussed in the next section.

## 2.2 The Layer of Actors, Actions and Actands

Transitions which are performed by someone are distinguished and called Actions. Actions are presented through a couple of tuples, the predicated thing of the first of which shows the performing Actor and of the second one the transition

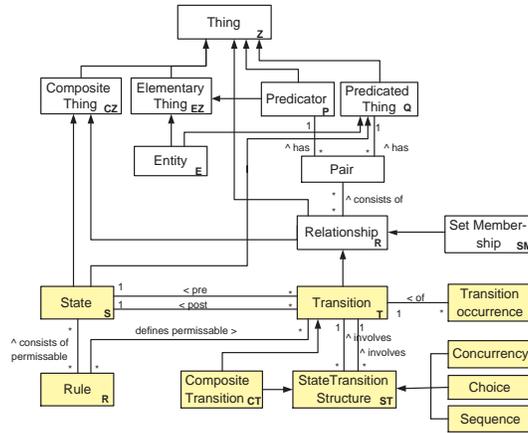


Fig. 2. Frisco - Fundamental Layer

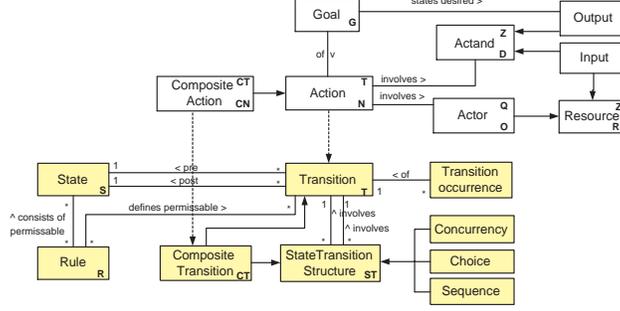
he/she is performing. The predicates used for describing this are the primitives performing and performed-by, correspondingly. The things involved in the input and the output states of an action, and which are not actors for that action, are called **Actands**. The input actands for an action (i.e. the actands from the input state) together with the actors are the **Resources** for that action. Also using the primitive is-context, some of the input actands can be predicated, in order to define the **Action context**. In a similar way the **Goal** of an action can be defined by intentionally stating the desired output state.

The formal definitions for this layer are reprinted in table 3 in the Appendix. The meta-model is drawn in figure 3. The transition part of the Fundamental layer, as closely related to the concepts defined here, is reprinted in the bottom of the figure. The isa relationships between the different layers are indicated by dotted arrows. Furthermore, the basic concept from the fundamental layer on which a concept is built on is indicated by giving the name of the set in the upper right corner of the rectangle of the class.

### 3 Results from the analysis of Frisco during the construction of the meta-model

During the process of building the meta-model we identified some minor problems within the formal definitions of Frisco. A discussion of them as well as suggestions for how to solve them are provided below.

The idea of defining State Transition Structure and Composite Transition concepts is to be able to build up complex transitions. However, the necessary recursive element for achieving this and assumed in the natural language description of the framework is missing from the formal definitions. This makes it unclear how complex transitions are actually built up. In the meta-model in



**Fig. 3.** Frisco - Actor, Action and Actands layer

figure 2, which is based on the formal definitions and not the textual description of the framework, this defect can be seen through the absence of a recursive relation. The relationship *involves* between State Transition Structure and Transition drawn twice in order to make explicit the fact that a State Transition Structure is built on two Transitions, does not capture the necessary recursion.

One way to solve this and to introduce recursion would be to define a State Transition Structure to be a Transition. This would not though work for Choice, where a unique post-state is not defined. Besides, it would make the definition of Composite Transition unnecessary. An alternative and more successful way would then be to include recursion within the definition of State Transition Structure. This is done by relaxing the existing assumption of  $t_x$  and  $t_y$  on being just Transitions into being Transitions or State Transition Structures, i.e. the following definition adjustment to the definition from table 2 is proposed.

**Def adj 1** If  $t_x : s_1 \rightarrow s_2, t_y : s_3 \rightarrow s_4$  are transitions or state transition structures then the following state transition structures exists

- Sequence:**  $sequ(t_x, t_y)$  if  $s_3 \subseteq s_2$
- Choice:**  $choice(t_x, t_y)$  if  $s_1 \cap s_3 \neq \emptyset$
- Concurrency:**  $concur(t_x, t_y)$  if  $s_1 \cap s_3 = \emptyset$

In the meta-model this change will be depicted by changing the range for one of the *involves* relationship from the current Transition to be State Transition Structure. The cardinality constraints of both *involves* relationship have to be adjusted, as it is not longer known in advance how many of the involved in the State Transition Structures elements are Transitions and how many are other State Transition Structures. This is depicted in figure 4a. More complicated ways for capturing this as well as a bit more of the semantics are available. However, even the simple solution we are proposing here exemplifies well enough how a meta-model can facilitate the analysis of a framework and how the model evolves to capture its changes.

Furthermore, going back to the concept of Transition Occurrence (as we've promised in the previous section), we note that interestingly, but not surprisingly,

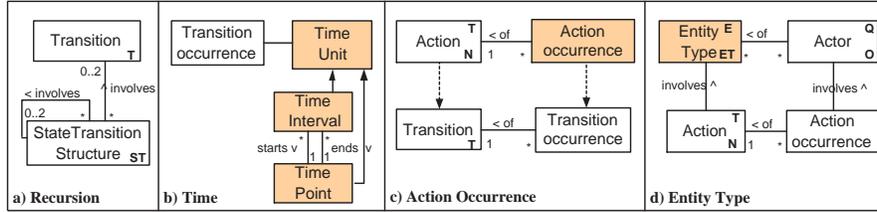


Fig. 4. Frisco - suggested improvements

after its formal definition in [1] an elaborated discussion on the formalisation of the time concept was given and how a relative ordering between transition occurrences could be done. However, even if discussed, the concept of absolute time was not formalised. Therefore, we propose the following formal definitions for Time Point, Time Interval and Time Unit concepts (for a detailed analysis of the time concept, please refer to [9]).

**Def 2** Let  $\mathcal{TP} = \{tp | tp \in \mathcal{EZ}\}$  denote the set of time points for which the following holds:

If  $tp_1 \in \mathcal{TP}$  and  $tp_2 \in \mathcal{TP}$  and  $tp_1 \neq tp_2$  then  $tp_1 < tp_2$  or  $tp_2 < tp_1$   
If  $tp_1 < tp_2$  and  $tp_2 < tp_3$  then  $tp_1 < tp_3$ , where  $tp_1, tp_2, tp_3 \in \mathcal{TP}$

**Def 3** Let  $\mathcal{TI} = \{\langle tp_s, tp_e \rangle | tp_s, tp_e \in \mathcal{TP} \wedge tp_s < tp_e\}$  denote the set of time intervals, where  $tp_s$  is called the start time point of an interval and  $tp_e$  is called the end of the interval.

**Def 4** Let  $\mathcal{TU} = \mathcal{TP} \cup \mathcal{TI}$  denote the set of time units.

After have defined a set of Time Units, it is now natural to extend the definition of Transition Occurrence with the notion of time. The following definition adjustment allows us to express not only that a transition has occurred, but also the time unit at which it occurred. The changes in the meta-model are shown in figure 4b. The distinction between a time point at which a transition occurs versus a time interval spanning over the occurrence of a composite transition, in the definition below, justifies the necessity of introducing both time points and time intervals.

**Def adj 5** A transition  $t : s_1 \rightarrow s_2$  is enabled to occur in state  $s$  if  $s_1 \subseteq s$ .

If a transition  $t : s_1 \rightarrow s_2$  occurs in state  $s$ , then  $s$  is changed to the new state  $s' = (s \setminus s_1) \cup s_2$ .

The occurrence of a transition  $t : s_1 \Rightarrow s_2$  in state  $s$  leading to state  $s'$  at time unit  $tu$  is denoted as  $occ(t, tu) : s \rightarrow s'$ . When the occurrence is of a composite transition the time unit  $tu$  is a time interval, otherwise it is a time point.

We are now turning our attention into the Action concept of the Actor, Action and Actands layer and how it builds on the Transaction concept. At the moment,

the Transition Occurrence concept from the Fundamental layer is used to capture both the fact that a transition occurs, as well as the fact that an action is performed. Graphically, it can be pointed out that the symmetry in figure 3 existing between the concepts Transition-Action and Composite Transition - Composite Action (from the Fundamental and Actor, Action, Actand layers correspondingly) is lacking for the concept Transition Occurrence. In order to make explicit the occurrences of the actions we suggest the introduction of an Action Occurrence concept (see figure 4c). Such a concept will naturally be based on the Transition Occurrence concept, which is indicated by the intra-layers isa relationship in the figure. In addition to bring explicitness and clarity to the model, this change allows for an independent consideration of the Actor, Action and Actands layer from the Fundamental layer. It also clearly establishes the Fundamental layer as a supporting layer which we also believe was its initial purpose.

Recall furthermore, that the difference between an Action and a Transition is that an action is performed by an actor. According to the formal definitions and to the meta-model in figure 3 when defining an action its actor is explicitly specified. However, having now introduced the concept of Action Occurrence, it would be far more natural to specify the actual performer of an action first when it has occurred. Furthermore, instead of specifying an Actor for the performance of an action it is rather natural to specify the potential performers for the action. We also find the limitation that for the moment an action can only be specified to be performed by an individual actor as quite unnecessary, especially as for the most actions (e.g., composite actions) the opposite is valid. In order to make the definition of an action more general, by allowing a whole group of actors to be specified as potential performers of it, we introduce the concept of **Entity Type**. This concept is actually used in some examples of the framework already. However it is not formally separated, but occurs as a part of the Type concept and even then it is not a part of its formal definition, but appears only in its exemplification. Whether this name is the most suitable one can be discussed. However, as we try to stay as close to the framework as possible we decide to keep it for now.

**Def 6** Let  $\mathcal{ET} = 2^{\mathcal{E}}$  denote the set of entity types.

Now, the definition of action is changed to specify an Entity Type as the performer of an action, while the fact that a specific actor has performed an action is captured in the definition for action occurrence adjusted below. The resulting change in the meta-model is shown in figure 4d.

**Def adj 7** Let  $\mathcal{N}$  denote the set of all actions.  $\mathcal{N} = \{t \in \mathcal{T} | \exists q \in \mathcal{ET}, \exists r \in \mathcal{R}[r = \{\langle q, performing \rangle, \langle t, performed\_by \rangle\} \wedge q \in Prestateof(t)]\}$

As argued above, we now give a separate definition for the concept of action occurrence.

**Def 8** An action  $n : s_1 \rightarrow s_2$  is enabled to be performed by an actor  $a$  in state  $s$  if  $s_1 \subseteq s$  and  $a \in Actorof(n)$ . If an action  $n : s_1 \rightarrow s_2$  is performed in state  $s$ , then  $s$  is

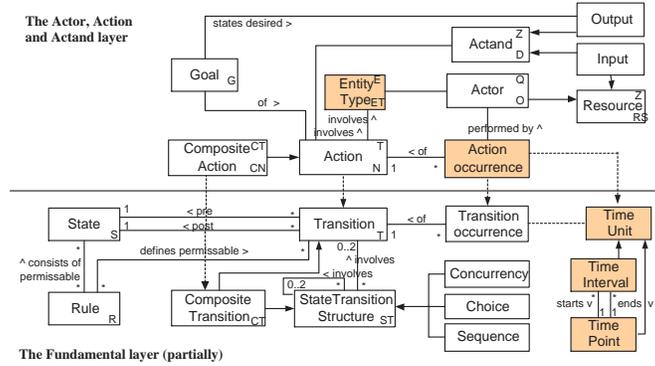


Fig. 5. Frisco - an extended meta-model

changed to the new state  $s' = (s \setminus s_1) \cup s_2$ .

The performance of an action  $n : s_1 \Rightarrow s_2$  by actor  $a$  in state  $s$  leading to state  $s'$  at time unit  $tu$  is denoted as  $occ(n, a, tu) : s \rightarrow s'$ .

This definition invokes the function  $Actorof$ , which takes as input an action and returns its performer, who is generally defined as a predicated thing. In order to unify it with the adjusted definition of an action, where the performer is an entity type, the following modification is suggested:

**Def adj 9** Let  $Actorof : \mathcal{N} \rightarrow 2^{\mathcal{E}}$  be a function determining the entity types specified as performers of an action, where  $Actorof(n) = \{q \in 2^{\mathcal{E}} \mid q \in Prestateof(n) \wedge \exists r \in \mathcal{R}[r = \{\langle q, performing \rangle, \langle n, performed\_by \rangle\}]\}$

The meta-model in figure 5 is drawn to summarize the results of the analysis and present the suggested improvements. The introduced concepts are indicated by shading the corresponding classes.

## 4 Conclusions and Further Work

In this work we focused on the construction of a meta-model for the Framework of Information Systems Concepts (Frisco). The initial results, i.e., the development of a meta-model for the first two layers of the framework was presented here and as a modelling language the notation of UML class diagrams was used. We believe this diagrammatical way of presenting a well formalised system of concepts will serve as a communication aid facilitating the dissemination of it.

Furthermore, we have demonstrated how the analysis provided through the meta-model development can be used in the further development of Frisco. This was done by in detail presenting the ambiguities identified and the amendments suggested to resolve them.

During the work Frisco was analysed in isolation. The next and even more interesting step will be to use the proposed meta-model for analysing Frisco across other alternative attempts existing in the area. In particular, the representatives of two rather contrasting approaches are of interest:

- The closely related, i.e., top-down developed and well formalised, frameworks or ontologies such as BWW, EO, TOVE are interesting to compare to Frisco so that overlap and discrepancies are outlined. Such an analysis would not only position the frameworks relative each other but it could also be beneficial for the potential development of each one of them (according to the technique of meta-model comparison as demonstrated in [7]).
- The pragmatic, bottom-up developed attempts like UEMML [10] and the framework proposed by Söderström et al. [11]. Such an analysis could initially be provided for surveying the similarities and differences between the approaches (and be beneficial for works like the one presented in [12]). Furthermore, mapping these non-formalised frameworks to the well-formalised Frisco (in line with the work presented in [13]) would also give clear semantics to them.

Common for all the attempts mentioned above is that they were all developed addressing the interoperability issues and that they are all supposed to be used for facilitating communication (between people, organisations and tools). Then an interesting research direction is to investigate the “interoperability” of the frameworks themselves.

## References

1. Falkenberg, E.D., et al.: FRISCO - A Framework of Information System Concepts. Technical Report ISBN 3-901882-01-4, IFIP (1998) <http://www.wi.leidenuniv.nl/~verrynst/fri-full-7.pdf>.
2. Wand, Y., Weber, R.: An ontological model on an information system. *IEEE Trans. Software Eng.* **16** (1990) 1281–1291
3. Uschold, M., Gruninger, M.: *Ontologies: Principles, methods and applications*. *Knowledge Engineering Review* **11** (1996)
4. OpenCyc.org: Opencyc 0.7b. Technical report, Cycorp (2004) <http://www.opencyc.org/doc/>.
5. Enterprise Integration Laboratory, U.o.T.: TOVE (TOronto Virtual Enterprise) Ontology Project. Accessed November 2004 from <http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html> (2004)
6. Uschold, M., King, M., Moralee, S., Zorgios, Y.: *The Enterprise Ontology*. Technical report, AIAI, The University of Edinburgh (1997) <http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html>.
7. Rosemann, M., Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. *Information Systems* **27** (2002) 75–91

8. OMG: UML 2.0 Superstructure Specification. OMG Draft Adopted Specification, ptc/03-08-02 (2003) <http://www.omg.org/cgi-bin/doc?ptc/2003-08-02>.
9. Allen, J.: Towards a general theory of action and time. *Artificial Intelligence* **23** (1984) 123–154
10. Panetto, H., et al.: A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. In: Proc. of the 11th IFAC INCOM2004 Symposium, Bahia, Brazil (2004)
11. Söderström, E., et al.: Towards a framework for comparing process modelling languages. In: Proc. of the 14th Int. Conf. on Advanced Inf. Systems Engineering (CAiSE). Volume 2348 of LNCS., Springer (2002)
12. Rukanova, B., Slooten, K., Stegwee, R.: Business Process Requirements, Modeling Technique, and Standard: How to identify interoperability gaps on a process level. In: Pre-proc. of the 1th Int. Conf. on Interoperability of Enterprise Software and Applications (INTEROP-ESA2005). (2005) 13–25
13. Wohed, P., Andersson, B.: (Reconciliation of two Business Modelling Frameworks) to be presented at CAiSE 2005 Forum, to be held in Porto, Portugal, June, 2005.

## Appendix

Concept	Set	Definition	Description
Thing	$\mathcal{Z}$	primitive	Any part of a conception of a domain
Relationship	$\mathcal{R}$	$\mathcal{R} = \{r \in \mathcal{Z}   r \subseteq \mathcal{U} \wedge \mathcal{U} = \{\langle q, p \rangle   q, p \in \mathcal{Z}\} \wedge 1 =  r  < \infty\}$	A thing composed of several predicted things, each one associated with one predictor characterizing the role of the predicted thing within the relationship
Predictor	$\mathcal{P}$	$\mathcal{P} = \{p \in \mathcal{Z}   \exists u \in \mathcal{U}, q \in \mathcal{Z} [u = \langle q, p \rangle]\}$	An atomic thing used to characterize or qualify other things
Predicted thing	$\mathcal{Q}$	$\mathcal{Q} = \{q \in \mathcal{Z}   \exists u \in \mathcal{U}, p \in \mathcal{P} [u = \langle q, p \rangle]\}$	The thing being characterized or qualified by a predictor
Set membership	$\mathcal{SM}$	$\mathcal{SM} = \{sm \in \mathcal{R}   sm = \{\langle q_1, has - element \rangle, \langle q_2, is - element - of \rangle\} \wedge q_1, q_2 \in \mathcal{Q} \wedge q_1 \neq q_2\}$	A binary relationship between a thing (the set) characterized by a special predictor called 'has-element' and another thing characterized by a special predictor called 'is-element-of'
Elementary thing	$\mathcal{EZ}$	$\mathcal{EZ} = \{ez \in \mathcal{Z} \setminus \mathcal{R}   \neg \exists u \in \mathcal{U} [u = \langle ez, has - element \rangle]\}$	A thing not being a relationship and not being characterized by the predictor 'has-element'
Entity	$\mathcal{E}$	$\mathcal{E} = \mathcal{EZ} \cap \mathcal{Q}$	A predicted thing which is an elementary thing
Composite thing	$\mathcal{CZ}$	$\mathcal{CZ} = \mathcal{Z} \setminus \mathcal{EZ}$	A non-elementary thing

**Table 1.** Frisco - The Fundamental Layer (based on the primitives: Thing, has-element, is-element-of)

Concept	Set	Definition	Description
Transition	$\mathcal{T}$	$\mathcal{T} = \{t \in \mathcal{R}   \exists s_b, s_a \in \mathcal{CZ} [t = \{\langle s_b, before \rangle, \langle s_a, after \rangle\} \wedge s_b \neq s_a]\}$	A binary relationship between two different composite things, called pre-state and post-state of the transition
State	$\mathcal{S}$	$\mathcal{S} = \{s \in \mathcal{CZ} \setminus \mathcal{T}   \exists t \in \mathcal{T} [s \in Predthingin(t)^3]\}$	A composite thing involved as a pre-state or post-state in a transition. No element of a state may be a transition itself.
State-transition structure	$\mathcal{ST}$	If $t_x : s_1 \rightarrow s_2, t_y : s_3 \rightarrow s_4$ are transitions then <b>Sequence:</b> $sequ(t_x, t_y)$ if $s_3 \subseteq s_2$ <b>Choice:</b> $choice(t_x, t_y)$ if $s_1 \cap s_3 \neq \emptyset$ <b>Concurrency:</b> $concur(t_x, t_y)$ if $s_1 \cap s_3 = \emptyset$	A transition which is either a Sequence, a Choice or a Concurrency
Composite transition	$\mathcal{CT}$	$\mathcal{CT} = \mathcal{ST} \cap \mathcal{T}$	A state transition structure with unique pre- and post- states
Transition occurrence		A transition $t : s_1 \rightarrow s_2$ is enabled to occur in state $s$ if $s_1 \subseteq s$ If a transition $t : s_1 \rightarrow s_2$ occurs in state $s$ , then $s$ is changed to the new state $s' = (s \setminus s_1) \cup s_2$ The occurrence of a transition $t : s_1 \Rightarrow s_2$ in state $s$ leading to state $s'$ is denoted as $occ(t) : s \rightarrow s'$	A specific occurrence of a transition
Type	$\mathcal{TY}$	$\mathcal{TY} \subseteq 2^{(\mathcal{Q} \cup \mathcal{CZ})}$	A specific characterization applying to all things of that type
Population	$\mathcal{PO}$	$\mathcal{PO} = \{po \in 2^{(\mathcal{Q} \cup \mathcal{CZ})}   \exists ty \in \mathcal{TY} [po \subseteq ty]\}$	
Instance	$\mathcal{IN}$	$\mathcal{IN} = \{in \in \mathcal{Q} \cup \mathcal{CZ}   \exists ty \in \mathcal{TY} [in \in ty]\}$	
Rule	$\mathcal{R}$	$\mathcal{R} = 2^{\mathcal{S}} \cup 2^{\mathcal{T}}$	A set of permissible states and transitions in a specific context

**Table 2.** Frisco - The Fundamental Layer, continuation (Primitives: before, after)

3. Let  $Predthingin: \mathcal{R} \rightarrow 2^{\mathcal{Q}}$  be a function from relationships to sets of predicted things, where  $Predthingin(r) = \{q \in \mathcal{Q} | \exists p \in \mathcal{P} [\langle q, p \rangle \in r]\}$ .

Concept	Set	Definition	Description
Action	$\mathcal{N}$	$\mathcal{N} = \{t \in \mathcal{T}   \exists q \in \mathcal{Q}, \exists r \in \mathcal{R}[r = \{\langle q, performing \rangle, \langle t, performed - by \rangle\} \wedge q \in Prestateof(t)^4]\}$	A transition involving a non-empty set of actors
Composite action	$\mathcal{CN}$	$\mathcal{CN} = \mathcal{ST} \cap \mathcal{N}$	A composite transition with the same condition as applying for the notion of action
Actor	$\mathcal{O}$	$\mathcal{O} = \{q \in \mathcal{Q}   \exists n \in \mathcal{N}[q \in Actorof(n)^5]\}$	A thing able to cause transition
Actand	$\mathcal{D}$	$\mathcal{D} = \{z \in \mathcal{Z}   \exists n \in \mathcal{N}[z \in Inputof(n) \vee z \in Outputof(n)^6]\}$	A thing involved in the pre-state or post-state of an action and not considered as an actor of that action
Resource	$\mathcal{RS}$	$\mathcal{RS} = \{z \in \mathcal{Z}   \exists n \in \mathcal{N}[z \in Prestateof(n)]\}$	The union of the set of actors and the set of input actands of that action
Action context	$\mathcal{X}$	$\mathcal{X} = \{d \in \mathcal{D}   \exists n \in \mathcal{N}, \exists r \in \mathcal{R}[r = \{\langle d, is - context \rangle\} \wedge d \in Inputof(n)^7]\}$	An optional part of the pre-state of an action qualifying the context or situation in which that action is performed, and determining or modifying at least one of its output actands.
Goal	$\mathcal{G}$	$\mathcal{G} = \{d \in \mathcal{D}   \exists n \in \mathcal{N}, \exists o \in \mathcal{O}, \exists r \in \mathcal{R}[r = \{\langle d, pursued - by \rangle, \langle o, pursuing \rangle\} \wedge d \in Inputof(n) \wedge o \in Actorof(n)]\}$	A special input actand of that action, pursued by the actors of that action and stating the desired output state intentionally

**Table 3.** Frisco - The Layer of Actors, Actions, and Actands (Primitives: performing, performed-by, is-context, pursued-by, pursuing)

- 
4. Let  $Prestateof: \mathcal{T} \rightarrow \mathcal{S}$  be a function from transitions to states, where  $Prestateof(t: s_b \Rightarrow s_a) = s_b$  denotes the (pre-)state before the transition  $t$ . Let  $Poststateof: \mathcal{T} \rightarrow \mathcal{S}$  be a function from transitions to states, where  $Poststateof(t: s_b \Rightarrow s_a) = s_a$  denotes the (post-)state after the transition  $t$ .
5. Let  $Actorof: \mathcal{N} \rightarrow 2^{\mathcal{Q}}$  be a function determining the actors performing an action, where  $Actorof(n) = \{q \in \mathcal{Q} | q \in Prestateof(n) \wedge \exists r \in \mathcal{R}[r = \{\langle q, performing \rangle, \langle n, performed - by \rangle\}]\}$ .
6. Let  $Outputof: \mathcal{N} \rightarrow 2^{\mathcal{Z}}$  be a function determining the output actands of an action, where  $Outputof(n) = \{z \in Poststateof(n) | \neg(z \in Actorof(n))\}$ . Let  $Inputof: \mathcal{N} \rightarrow 2^{\mathcal{Z}}$  be a function determining the input actands of an action, where  $Inputof(n) = \{z \in Prestateof(n) | \neg(z \in Actorof(n))\}$ .