Reconciliation of two Business Modelling Frameworks

Petia Wohed[1] and Birger Andersson[2]

[1] CRAN, Université Henri Poincaré, Nancy 1/CNRS
BP239, 54506 Vandoeuvre les Nancy, France
petia.wohed@cran.uhp-nancy.fr
[2] DSV, Stockholm University/Royal Institute of Technology
Forum 100, SE-164 40 Kista, Sweden
ba@dsv.su.se

Abstract. Addressed in this paper is the problem of conceptual heterogeneity within the field of information systems. Two frameworks, Frisco and Söderström, each reflecting this heterogeneity, are presented. They are analysed and an reconciliation attempt of them is provided. The reconciliation points at some strengths and weaknesses in each framework.

1 Introduction

Frameworks like Frisco [1] and BWW [2] were developed during the last decade to adress the terminological fuzziness characterizing the field of information systems. By proposing a coherent system of concepts their common goal was to provide a uniform terminology which could be used to reduce misunderstandings and ambiguities. In the same time period, also addressing terminological heterogeneity, frameworks like those of Söderström [3] and UEML [4] were proposed. They were primarily developed for facilitating structured and unbiased analysis when comparing business process modelling languages.

While Frisco and BWW are developed top-down, are formal and founded on solid philosophical foundations, the frameworks of Söderström and UEML are developed bottom-up, are informal and are grounded on practical rather than theoretical work. A common feature of these rather orthogonal approaches, however, is that the achievement of their goals would facilitate the development of interoperable information systems.

Currently, none of these frameworks has yet received the critical mass of attention and deployment for attaining their goals. Criticizing Frisco and BWW for being too theoretical, and Söderström and UEML for not being formalized and therefore fuzzy, are fair concerns and this criticism has been the motivation for the work reported in this paper.

In order to take advantage of the strengths and mitigate the weaknesses, a reconciliation of two of these frameworks, Frisco and Söderström's, is proposed. By combining the theoreticians' with practitioners' work, we hope to achieve a good mixture of sound theoretical grounding with user-friendliness, thereby increasing the understandability and applicability of the result. This paper proceeds by first presenting both frameworks followed by an analysis and an attempt to reconcile them.

Proceedings of the CAiSE'05 Forum - O. Belo, J. Eder, J. Falcão e Cunha, O. Pastor (Eds.)

2 Söderström's Framework

This section is opened by presentation of the framework taken from [3], followed by discussion of its ambiguities. The meta-model is reprinted in figure 1.

> "Most process modelling languages include at least four basic concepts: time point, activity, state and event, [..]. Intuitively, a *time point* is an instant in time, not further decomposable. An *activity* is a performance of some sort, possibly changing some thing's *state*, i.e. its set of properties. An *event* is a noteworthy occurrence. Usually, one is interested in particular events associated with changes of state, i.e. activities are involved in some way.
>
> – Events can either record a certain point in time (*time point event*) or record the time between two time points (*time duration events*).
> – Events can either record the start of an activity (*pre-activity events*) or record the end of an activity (*post-activity events*).
> – Events may record the change of a state (*state change events*) or not.
>
> A process is modelled [..] as a structure of logical dependencies between activities. These activities use one or more resources as input, and produces one or more resources as output. One specific type of resource, the role, is regarded to be responsible for that one or more activities will be performed.
>
> The execution of a process is regarded to be a time-bound series of events, caused by an actor. These events may result in a state change for a resource. An actor who causes an event always has an intention with his/her actions to achieve a specific state for a resource. The state can be either different from the resource's current state, or it can be the same state as the current one.
>
> [..] An event always occurs at certain time point or between two time points on a certain location."

Furthermore, the model is divided into two levels: a type level, containing activity, resource, role, activity dependency and process; and an instance level containing event, state, actor and temporal dependence.

The pronouns: where, who, how, what, and why have been added into the framework to enhance its readability. However, as they do not bring any additional semantic we have left them out of our discussion.

From the description provided above the following questions arise.

– How are the entities Process and Activity Dependency related to each other? A process is composed of a number of activities, strictly defining the order between them, while Activity Dependency is the entity aimed for capturing in pairs the ordering between activities.

– What in the framework is referred to as *type* and *instance level* is what we associate to Fowler's *knowledge* and *operational level* [5]: operational level aimed for capturing the day to day events of a domain; and knowledge level

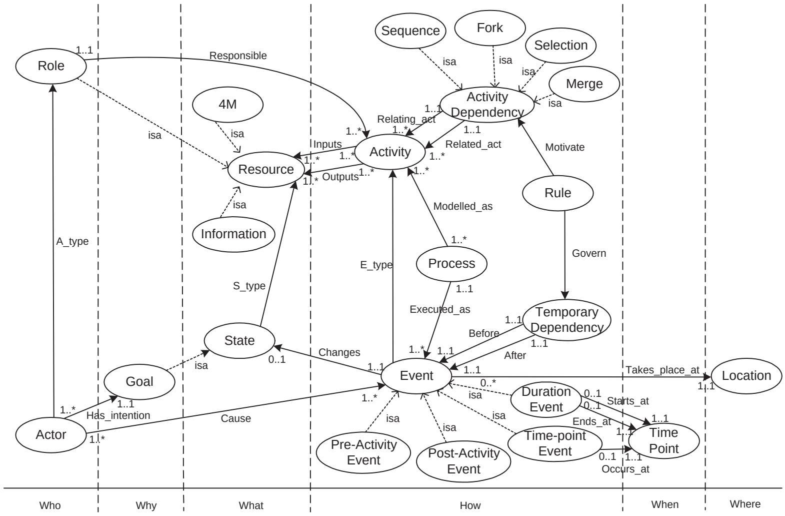

Fig. 1. Söderström's meta-model

capturing general rules governing in the domain and the events occurring within it. A consequence of Fowler's approach, is that many of the entities at the operational level tends to more generally be described through (i.e., mapped to) corresponding entities at the knowledge level. For instance, an Actor performing an Event (both modelled at the operational level) are through the relationships A_type and E_type mapped to Role responsible for an Activity at the knowledge level. For instance, the Event *1 April 2004 Eva books flight tickets* performed by *Eva* (an Actor) in her Role of *secretary* implying the responsibility of *tickets booking* Activity. Likewise, the Process *Travel planning and accomplishment* (at knowledge level) could at operational level be exemplified by *Ann's travel planning and trip to London.* However, there is not any corresponding entity to Process at operational level. The question arising is: Why are some entities at operational level missing corresponding entities at knowledge level?

– According to the mapping constraints one role is responsible for the performance of an activity. However, some activities are allowed to be performed by the holders of different roles. This would in the model be captured through the Input relationship between Resource (as a generalization of Role) and Activity, which is not an intuitive solution. Furthermore, the isa relationship between Role and Resource is not entirely clear. Applying the operational/knowledge level thinking introduced by Fowler, this isa relationship would transfer to an isa relationship between Actor and State entities in the operational level. As this generalization/specialization does not make any sense it is naturally not present in the framework. However, this ambiguity confirms the need for further clarification of the concepts in the framework.

– The notion of non state changing events is somewhat ambiguous, as one could claim that events are capturing state changing phenomena.

– Rule, 4M, and Information concepts are currently not described in the natural language description of the framework.

3 Frisco

In this section a brief introduction to the relevant parts of Frisco [1] is given. A UML meta-model is drawn in figure 2.

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.

Furthermore, a special kind of relationship, consisting of a couple of tuples, called Transition, is introduced. The predicators in the tuples within a transition are the primitives before and after and the predicated things are composite things, so called States. Complex transitions can be build up through Sequence, Choice and Concurrency 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.

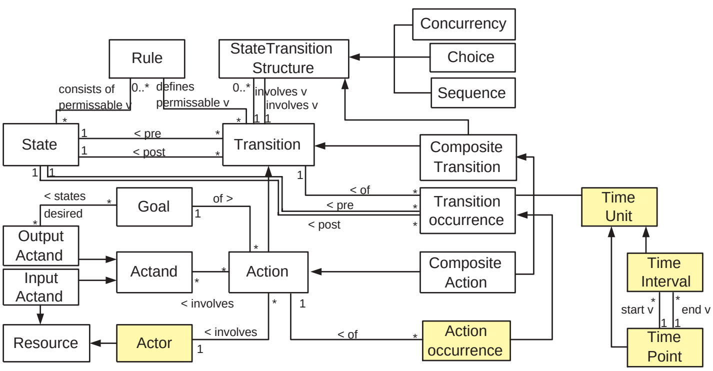

Fig. 2. Frisco - a partial metamodel

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 he/she is performing. The predicators 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.

During the analysis of Frisco, few ambiguities were discovered and some improvements suggested. These resulted in the definition of Time Point, Time Interval, Time Unit, Entity Type and Action Occurrence concepts (see the shaded classes in figure 2) Due to space limitation, for details and formal definitions the reader is referred to [6].

4 Mapping the Frameworks

The mapping of the frameworks we suggest is provided in table 1. Hence, the formal definitions in Frisco [1] and [6], provides a formalization of Söderström's framework.

Söderström	Frisco	Söderström	Frisco
Activity	Action	Event	Action Occurrence
Process	Composite Action	Duration Event (Process Occurrence)	Action Occurrence (of comp. action)
Resource	Resource Output Actand	Time-point Event	Action Occurrence (of non comp. action)
Resource	Input Actand	Pre-activity Event	–
Resource	Output Actand	Post-activity Event	–
Role	Entity Type	Actor	Actor
4M	–	State	State
Information	–	Goal	Goal
Activity Dependency	State Transition Structure	Temporal Dependency	–
Sequence	Sequence	Location	–
Fork	Concurrency	Time point	Time point
Select	Choice	–	Time interval
Merge	(Choice+Sequence)	–	Time unit
Rule	Rule	–	Action context

Table 1. Mapping of the frameworks

As indicated by the empty cells, there are concepts which can not be mapped directly. For instance the Location concept, as well as Pre- and Post-activity Event concepts are lacking in Frisco. We have also for the moment refrained mapping the concepts Information, 4M and Temporal dependencies from Söderström as we are lacking sufficient understanding of them. However, by providing a formal foundation for Söderström's framework, further elaboration and clarification of these concepts is facilitated.

The central concepts of Söderström's framework, i.e., the concepts Activity and Event are mapped into Frisco's Action and Action Occurrence, correspondingly. Process is mapped to Composite Action, which also implies certain changes to it. Namely, the fact that a Composite Action is defined as subtype to Action, forces the same semantic on the relationship between Process and Activity.

Furthermore, an Action Occurrence of a Composite Action, which is possible to express in Frisco, would actually correspond to a Process Occurrence. However, no such concept exists in Söderström. Instead, the distinction between Duration and Time-point Events indicating the time-span differences is made. This is also one of the differences captured by Action Occurrence of composite actions versus Action Occurrence of actions which are not composite. Therefore these concepts are mapped on each other as proposed in the table. Note, though, that this mapping forces the semantics of a composite action consisting of a number of actions, into Duration Event, which was not actually explicitly stated in Söderström. This implies also that the term Duration Event is unnatural and the term Process Occurrence, given in brackets, is suggested.

Moreover, Frisco has a more fine-grained conceptualization of resources specifying a Resource to be an Actor or an Input Actand (i.e., the object on which an action is performed) and distinguishing them from the Output Actands (i.e. the results from actions). As this distinction is not made in Söderström, the Resource concept of it is mapped into the union of Resource and Output Actand.

Finally, as a Role is specified to be responsible for the performance of an Activity, which corresponds to the Entity Type specified for the performance of an Action, Role and Entity Type are mapped to each other. Activity Dependency with its subclasses has been mapped into State Transition Structure with the corresponding subclasses. Also Söderström's Actor, State, Goal, Rule, and Time Point have directly been mapped into the corresponding Frisco concepts.

To visualize the result we propose a UML meta-model in figure 3. The classes are denoted by the terms from Söderström in the upper half of the rectangles and the terms from Frisco in the bottom half. Only the concepts supported by Frisco's formal definitions are presented.

From this meta-model it can be seen that the ambiguity of the relationship between Process and Activity Dependency have now been solved. Furthermore, the isa relationship from Role to Resource has been removed, hence the ambiguity around it solved. The Resource entity has been split into two entities, i.e., Input- and Output Actand entities. The Rule entity has got a clear semantic. The State entity has explicitly been related to several other concepts.

5 Conclusions

Concluding the discussion we would like to stress the following points. The distinction of Knowledge/Operational levels provided in Söderström forced some clarifying changes in this matter in Frisco [6]. As the level distinction is still not complete for all of Frisco's concepts and we believe that further work in this direction would bring additional benefits to the framework.

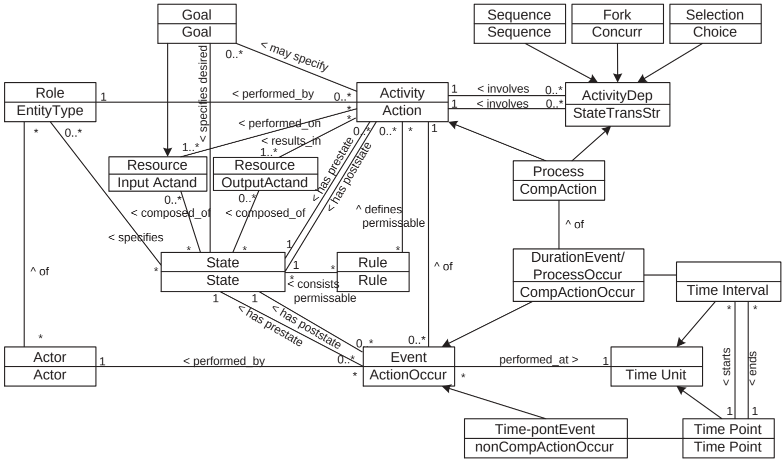

Fig. 3. Consolidated Model

Furthermore, formalizing Söderström's framework have resulted in clarification of it. However, not being able at this stage to capture all its concepts especially the various Event Types is a concern. The formalization proposed here is, however, a first step providing the necessary ground on which further conceptualization refinement can be made.

Finally, one of the major benefits of this integration is that its result, as a combination of a graphical with a formal representation, facilitates further cross-analysis of the approach with similar attempts in the area [2, 7].

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. 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)
4. 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)
5. Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley (1997)
6. Wohed, P., Andersson, B.: Unification of Terminological and Paradigmatic Differences in Two Ontologies. Technical report, DSV, SU/KTH (2005) `http://www.dsv.su.se/~petia/Publications/FS-TR05.pdf`.
7. Rosemann, M., Green, P.: Developing a meta model for the Bunge-Wand-Weber ontological constructs. Information Systems **27** (2002) 75–91