BPMN as a Communication Language for the Process- and Event-Oriented Perspectives in Fact-Oriented Conceptual Models

Peter Bollen

School of Business and Economics
Maastricht University, The Netherlands
Tel.: 0031-43-3883715; Fax: 0031-43-3884893
p.bollen@maastrichtuniversity.nl

Abstract. In this paper we will show how the OMG specification of BPMN (Business Process Modeling Notation) can be used to model the process- and event-oriented perspectives of an application subject area. We will illustrate how the fact-oriented conceptual models for the information-, process- and event perspectives can be used in a 'bottom-up' approach for creating a BPMN model in combination with other approaches, e.g. the use of a textual description. We will use the common doctor's office example as a running example in this article.

1 Introduction

The fact-oriented conceptual modeling approach enables analysts to precisely model the content of the data perspective for an application subject area as a set of fact types and population (transition) constraints and (parts of) the process-oriented perspective, mainly by using derivation rules (ORM [1], NIAM [2], CogNIAM [3]) or different types of state (transition) models [4, 5]. A further treatment of the process- and event perspectives in fact-oriented conceptual modeling, has been given in [5-10].

The development of OMG's SBVR into a standard for expressing business rules is an achievement for the fact-oriented modeling approach because it clearly embraces the idea that the only fact-encoding construct is the fact type and it clearly rejects any form of attribute as a main fact-encoding construct.

However, SBVR mainly covers the business rules in the information perspective, but it has no modeling provisions for modeling the business rules in the process- and behavioural perspectives [11-13]. From this the question arises which modeling language standard can be used in tandem with SBVR to express the business rules in the process- and behavioural pespectives.

The business process modeling notation (BPMN) v 1.1. [14] is an OMG standard that defines the modeling elements for creating business process models. BPMN was developed with the primary goal of providing a notation that would be readily understandable by all business users, from business analysts to technical developers that will have to implement the technology [15, p.1] and the need to translate 'original' business process models into 'execution' models.

In this paper we will illustrate how the Business Process Modeling Notation (BPMN) can be used to express the business rules in the process and behavioural perspectives of an enterprise and therefore, in combination with SBVR will allow business analyst to use the modeling results of the fact-oriented approach, that is characterized by it's clarity, user-validation mechanisms and expressiveness and express it in OMG's SBVR and BPMN standards [11].

In section 2 we will introduce the essential BPMN modeling constructs. In section 3 we will summarize the fact-oriented conceptual schema for the PDO case study as was derived in [16]. In section 4 we will illustrate how we can map a complete fact-oriented conceptual schema onto a BPMN model. We will use a popular fact-oriented notation for creating the models in the information perspective: Object-Role Modeling [1] extended with an explicit naming convention for fact types and constraints (as in [17]).

2 OMG's BPMN Standard for Business Process Models

The business process modeling notation (BPMN) v 1.1. [3] is an OMG standard that defines the modeling elements for creating business process models. The core modeling elements are defined on [14, pp. 18-20]. The BPMN basic concepts can be classified into *flow objects*, *connectors* and *artifacts* [15]. The flow objects and the sequence flows are the main modeling constructs that can be used to capture the essence of the models in the process-and behavioural perspectives. These modeling artifacts mainly provide the 'bridge' between the processes and the information that is needed to execute the process and the information that is the outcome of such a process execution. In figure 1 we have depicted the graphical representations of the most important BPMN modeling constructs.

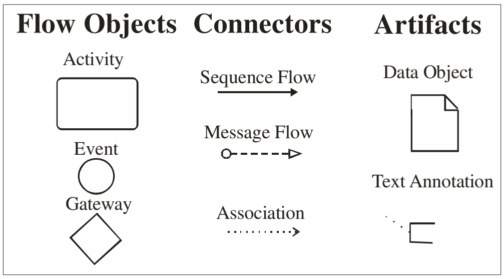

Fig. 1. Diagrammatic representation of most important BPMN modeling concepts

3 The Fact-Oriented Conceptual Schema for the PDO Case Study

The process- and event-oriented elements of the conceptual schema can now be expressed as BPMN *ad-hoc processes*. We will apply a combined 'bottom-up' and a 'case description' perspective for creating BPMN models for different aspects, e.g. 'business domain'- and 'implementation' views on the business processes. We will illustrate this mapping with the 'bench-mark' patient and doctor's office (PDO) case study as is described in [18].

3.1 The Information Model of the PDO Case Study

The ‘traditional’ ORM conceptual schema for the information perspective for the PDO case study is given in figure 2. In contemporary fact-oriented approaches, the domain ontology plays a dominant role, and is usually captured in some form of ‘list of definitions’ (e.g. see [6, 19, 20]). For the PDO domain ontology we refer to the list of definitions in an earlier contribution to this workshop [16].

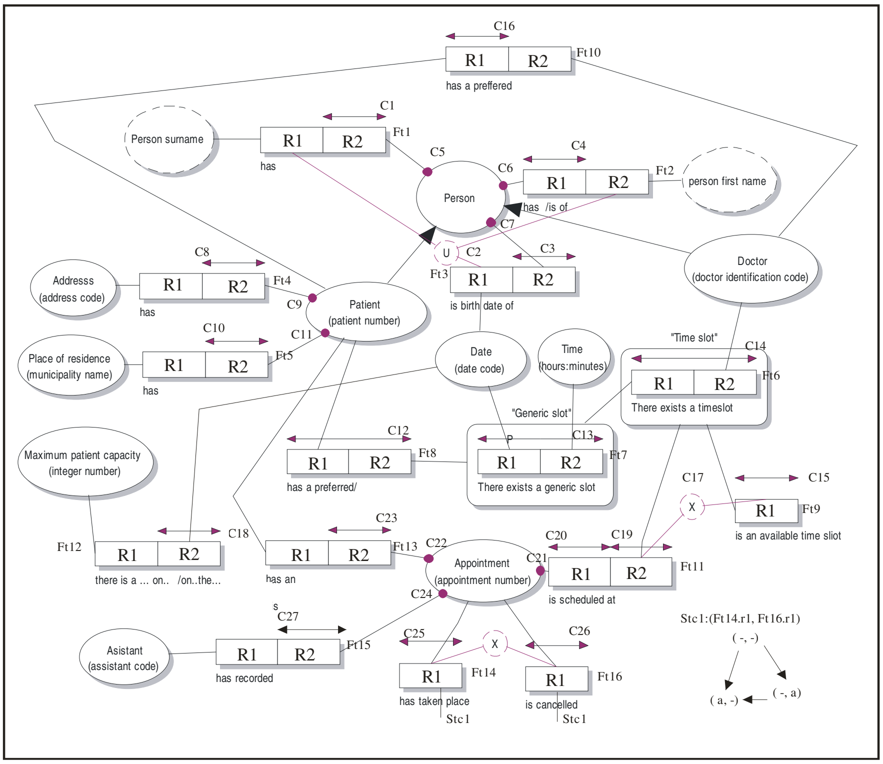

Fig. 2. (Excerpt from) list of definitions and fact type diagram for the PDO example

3.2 The Process Description of the PDO Case Study

The derivation and exchange rules in a fact-oriented conceptual schema provide a complete description of the semantics in the process-oriented perspective for a given application subject domain. The derivation rules create ‘new’ fact instances out of ‘old’ fact instances. Exchange rules, basically (de)populate the application’s information base by adding and/or deleting (atomic) fact instances. In this article we will only discuss a small (but significant) number of derivation-, exchange- and event rules from our PDO case study. For a complete overview we refer to [16].

Derivation rule *Dr1* derives the total number of patients that are registered in the practice at any point in time.

```
Dr1: Calculate total number of patients
IF EXT (Ft4) is empty
THEN total number of patients.R1:=0
ELSE  total number of patients.R1:=COUNT(EXT(Ft4))
ENDIF
```

In exchange rule *Ex9* an appointment can be added.

```
Ex9: Add appointment(arg1:date, arg2:time, arg3:doctor,arg4:patient, arg5:assistant)
IF [‘arg4’  in EXT(Ft5.R2) AND ‘arg1, arg2, arg3’ in Ft9.R1]
THEN CREATENEWAPPOINTNUMBER: [b]
        [( ADD an instance of Ft13 such that Ft13.R1=‘arg4’ AND Ft13.R2= b) ) AND
         (ADD an instance of Ft11 such that Ft11.R1= ‘b’ AND Ft11.R2:= ‘ arg1, arg2, arg3’)
        AND (ADD an instance of Ft15 such that Ft15.R2= ‘b’ AND Ft15.R1:= ‘ arg5’)]
ENDIF
```

We note that in the textual description of the PDO case study, trivial exchange rules are not mentioned explicitly. In our work-out of the case study in [16] we have added these exchange rules, thereby we have made assumptions about the extent in which instances of fact types are allowed to be inserted, updated or removed on their own.

3.3 The Event Description of the PDO Case Study

An event rule in a fact-oriented conceptual schema depicts under what conditions the occurrence of an event leads to the execution of one or more derivation- and/or exchange rules. In the PDO example, in event rule *ER2*, it is exactly stated what happens when an unknown patient wants to make an appointment.

```
ER2: Unknown patient requests for appointment
ON E1: Person wants to make a doctor’s appointment (arg1: first name, arg2: surname,
       arg3: date of birth)
IF [Dr2: Determine patient status (arg1= ‘E1.arg1’, arg2= ‘E1.arg2’, arg3= ‘
    E1.arg3’) = ‘Patient is not yet known’]
DO Ex1: Register patient (arg1: first name, arg2: surname, arg3: date of birth,
         arg4: address, arg5: place of residence)
[Impulse mapper: Ex1.arg1:= ‘E1.arg1’; Ex1.arg2:= ‘E1.arg2’; Ex1.arg3:= ‘E1.arg3’]
```

4 Mapping Fact-Oriented Conceptual Models to BPMN Models

In this section we will provide a mapping from the process- and event perspectives to the BPMN modeling constructs that were described in section 3 of this paper.

4.1 Mapping of the Fact-Oriented Models in the Process-Perspective to BPMN Ad-Hoc Process Standards

The first mapping we will provide maps the fact-oriented derivation- and exchange rules onto the ‘ad-hoc’ BPMN process model ‘modules’. The concept of derivation rule can be mapped as an BPMN activity. The fact type(s) that are input(s) to a derivation rule are encoded as (an) *input* data-object(s) and the fact type(s) that is (are) derived in the derivation rule will be encoded as (an) *output* data-object(s). These input- and output- data objects are connected to the activit(y)ies by means of (a) ‘directed’ association(s).

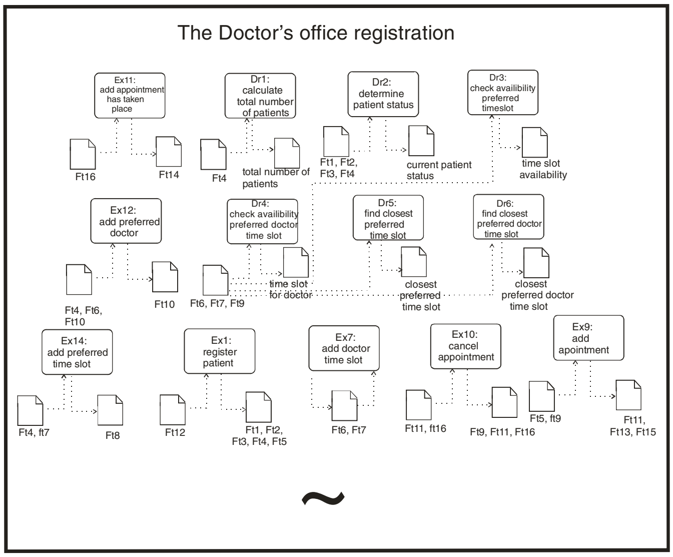
The Doctor's office registration
Ex11: add appointment has taken place
Dr1: calculate total number of patients
Dr2: determine patient status
Dr3: check availibility preferred timeslot
Ft16
Ft14
Ft4
total number of patients
Ft1, Ft2, Ft3, Ft4
current patient status
time slot availability
Ex12: add preferred doctor
Dr4: check availibility preferred doctor time slot
Dr5: find closest preferred time slot
Dr6: find closest preferred doctor time slot
Ft4, Ft6, Ft10
Ft10
Ft6, Ft7, Ft9
time slot for doctor
closest preferred time slot
closest preferred doctor time slot
Ex14: add preferred time slot
Ex1: register patient
Ex7: add doctor time slot
Ex10: cancel appointment
Ex9: add apointment
Ft4, ft7
Ft8
Ft12
Ft1, Ft2, Ft3, Ft4, Ft5
Ft6, Ft7
Ft11, ft16
Ft9, Ft11, Ft16
Ft5, ft9
Ft11, Ft13, Ft15

Fig. 3. Mapping of fact-oriented derivation- and exchange rules onto BPMN 'ad-hoc' process models (only graphical elements)

As a second mapping we will provide the mapping of exchange rules in the process perspective. The collection of exchange rules basically limits the playing field on how instances of a 'base- or asserted fact type' [21, p.99] can be added to or removed from the information base.

In fact-oriented process- and event models we can consider the 'fact-creating' activities as processes that can no longer be decomposed. This means that the mapping onto BPMN models is always on *a task* level. For a discussion on 'conceptual-process' types as 'fact-creating' activities we refer to [7, 17, 22, 23]. In figure 3 we will only display the graphical BPMN elements (and therefore leave out the attributes for the activity). As an example of how we can use the process attributes to capture the semantics in the process-oriented perspective we have given the attribute values for the process *dr1* in table 1 including the related *input sets*, *output sets* and *IOrules* attributes. The fact types that are used in a condition and/or expression in the derivation rules are contained in the BPMN input set of the activity. The resulting fact type that is connected to the ORM derivation rule will be mapped

Table 1. Attribute values for process dr1: calculate total number of patients

Attributes	**Prop/attribute**	**value for dr1**
Activity type		Task: *calculate total number of patients*
Input set	artifactInput	Data-object *Ft4*
Output set	artifactOutput	Data-object *total number of patients*
IOrules		IF EXT (Ft4) is empty
		THEN total number of patients.R1:=0
		ELSE total number of patients. R1:=COUNT(EXT(Ft4))
		ENDIF

as an output set. The ‘derivation algorithm’ in the ORM derivation rule will be mapped onto the activity’s *IOrules* attribute. Furthermore we will use the *assignment attribute* modeling construct to capture the (values of) the fact-oriented process argument and we will set the assign time attribute equal to ‘start’.

4.2 Mapping of the Fact-Oriented Models in the Event-Perspective to the BPMN

The third mapping that we will illustrate concerns the mapping of event rules onto the BPMN modeling constructs. In the event rules we will first map the event concept in an event. The impulse mapper construct encodes how a derivation/exchange rule argument is instantiated as a function of (amongst others) the value of the event argument. The impulse mapper will be modeled in BPMN as an assignment connected to the activity that is triggered by the event. In figure 4 and table 2 a transformation is shown for an event rule in which a condition must be satisfied before an event occurrence will lead to the execution of a derivation rule and/or exchange rule. In table 2 we have provided the attributes and values for the BPMN model elements that together represent the fact-oriented event rule 2 from the PDO example.

Table 2. Attribute values for event rule 2: unknown patient requests for appointment

Event Attribute		**value**
Event name		E1: Person wants to make doctor’s appointment
Event type		Start
Condition Attribute		**value**
Condition expression		Dr2: Determine patient status (arg1= ‘E1.arg1’, arg2= ‘E1.arg2’, arg3= ‘E1.arg3’) = ‘Patient is not yet known’
Activity Attribute		**value**
Activity type		Task: Ex1: register patient
Input set	data object 1 data object 2	Ft12 Total number of patients
properties	arg1 arg2 arg3 arg4 arg5	first name, surname, date of birth, address, place of residence
Output set	data object 1,2,3,4,5	Ft1, Ft2, Ft3, Ft4, Ft5
IOrules		IF [total number of patients< EXT(Ft12.R1) (where Ft12.R2=’arg6’)] THEN CREATENEWPATIENTNUMBER: [a][ADD an instance of FT1 to the ib where Ft1.R2:= ‘a’ and Ft1.R1:= ‘arg2’ AND ADD an instance of FT2 to the ib where Ft2.R1:= ‘a’ and Ft2.R2:= ‘arg1’ AND ADD an instance of FT3 to the ib where Ft3.R2:= ‘a’ and Ft3.R1:= ‘arg3’ AND ADD an instance of FT4 to the ib where Ft4.R2:= ‘a’ and Ft4.R1:= ‘arg4’ AND ADD an instance of FT5 to the ib where Ft5.R2:= ‘a’ and Ft5.R1:= ‘arg5’] ELSE DISPLAY (‘Maximum patient capacity has been reached’) ENDIF
Assignments		Ex1.arg1:= ‘E1.arg1’; Ex1.arg2:= ‘E1.arg2’; Ex1.arg3:= ‘E1.arg3’

5 The BPMN Example of the Combined Process and Event Perspectives of the PDO Case Study

In section 4 we have illustrated how the fact-oriented process- and event models can be mapped onto BPMN 'ad hoc' process model fragments. A specific 'process- flow' that can be implemented in an organizational setting, might, however require, more constraints in terms of process precedence requirements, parallelization, other than the ones that are already enforced, for example by process pre-conditions. In terms of business and cost considerations, one specific activity sequence might be favoured over other ones. This phenomena was illustrated in chapter 15 of Halpin and Morgan [21]. In this 'sampling methodology' example it was perfectly illustrated how business considerations, e.g. minimizing total costs, can lead to favour one 'Business-Process Design' over another one, not-withstanding that the underlying facts and activities are identical. In this section of this paper we will give one preferred 'Business-Process-Design' for the complete PDO case study [18] (for a complete description of the derivation-, exchange- and event-rules we refer to [16]), by embedding the activities, sequence flows, events that resulted from section 4 in the textual description of the PDO case study using, a specific set of consistent assumptions to create the 'specific business process design'.

In figure 4 we have used the *ad-hoc business process model* and *event-rules* from the fact-oriented conceptual model as 'building blocks' for the 'Business' BPMN model. We thereby, have added the behaviour of the external actors, by adding 'external actor' pools and additional message flows, that were not part of the fact-oriented process- and event models. If we inspect the BPMN model in figure 5, we can see that there are only three 'external' events that we need to distinguish in order to execute an instance of each of the three business processes: *person needs an appointment*, *person wants to cancel an appointment*, and *person has shown up at the appointment*. Furthermore, we have left out all the data-objects and their associations, as they were given in the 'ad-hoc' process model along the lines of the ones illustrated in figure 3.

By going through the textual description of the PDO case study, we will use the *application information grammar*, the *application process description* (which includes the *derivation-* and *exchange* rules) and the *application event description* as a starting point for determining the consistency and completeness of the BPMN description. The model modules that we have created by mapping the fact-oriented models onto an 'ad-hoc' BPMN process model, now have to be 'connected' in combination with the application's event description as follows:

1) Combine events under disjunct conditions into one event-gateway.
2) Check that every (relevant) fact-oriented process (from the 'ad-hoc') model, appears in at least one business process, in the business process diagram
3) Model activities that performed by the 'outside' actors in a separate pool

Now the business process diagram can be abstracted to create bird's eye views for different business user groups.

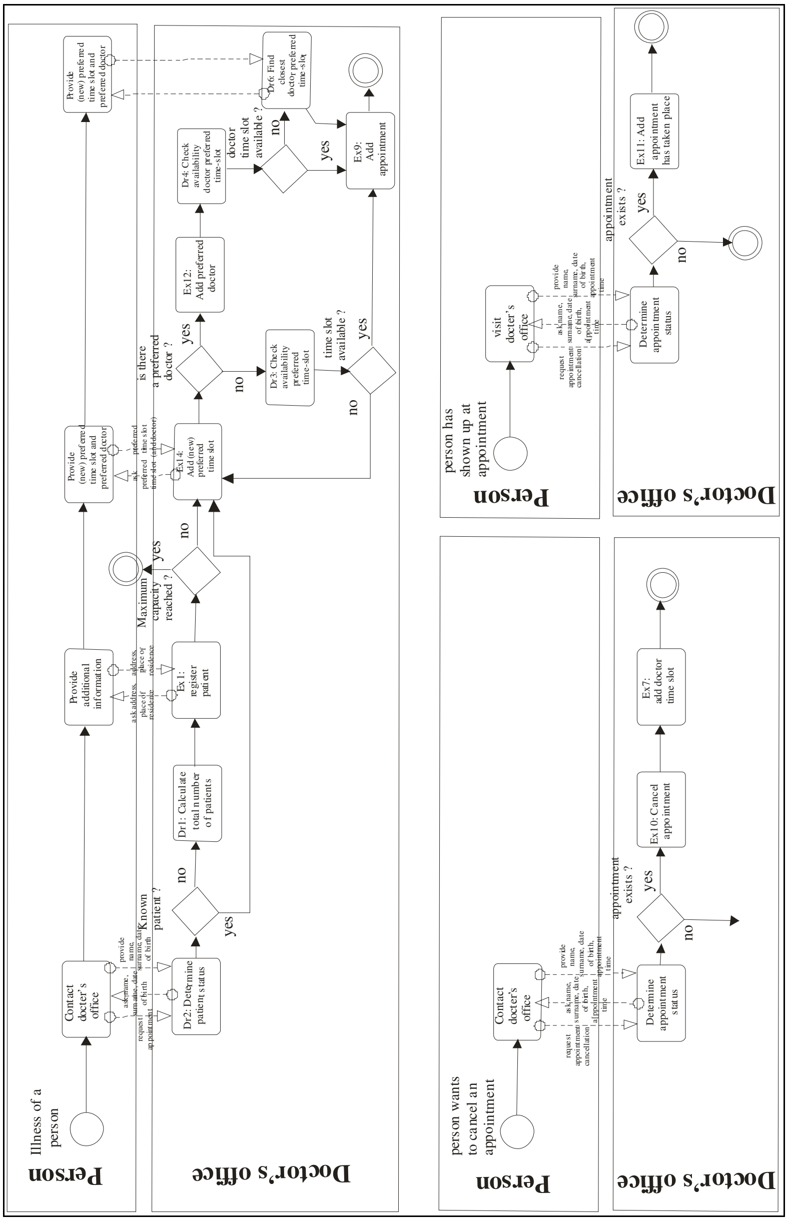
Person
Illness of a person
Contact docter's office
Provide additional information
Provide (new) preferred time slot and preferred doctor
Provide (new) preferred time slot and preferred doctor
Doctor's office
Dr2: Determine patient status
Known patient ?
yes
no
Dr1: Calculate total number of patients
Ex1: register patient
Maximum capacity reached ?
yes
no
Ex14: Add (new) preferred time slot
is there a preferred doctor ?
yes
no
Ex12: Add preferred doctor
Dr4: Check availability doctor preferred time-slot
doctor time slot available ?
yes
no
Dr6: Find closest doctor preferred time-slot
Dr3: Check availability preferred time-slot
time slot available ?
yes
no
Ex9: Add appointment
request appointment
ask name, surname, date of birth
provide name, surname, date of birth
ask address, place of residence
address, place of residence
ask preferred time slot
preferred time slot (and doctor)
Person
person wants to cancel an appointment
Contact docter's office
Doctor's office
Determine appointment status
appointment exists ?
yes
no
Ex10: Cancel appointment
Ex7: add doctor time slot
request appointment cancellation
ask name, surname, date of birth, appointment time
provide name, surname, date of birth, appointment time
Person
person has shown up at appointment
visit docter's office
Doctor's office
Determine appointment status
appointment exists ?
yes
no
Ex11: Add appointment has taken place

Fig. 4. Business Process Diagram

The advantage of mapping the fact-oriented models onto a BPMN process model is that the 'lowest-level' or the 'task-level' of activities is known. Furthermore all necessary, information processing activities are known, and will have to be covered as some element in a BPMN model. In addition, we will be able to model the 'process' and 'event' arguments with the messages that are sent between organizational objects in different pools. A summary of this BPMN-based business model creation process is given in table 3.

Table 3. Business model creation process using fact-orientation and BPMN

Stage	Input document	Resulting document
Stage 1: Create information model	Data use cases	Application information Model
Stage 2: Create process description	Application information model, Textual description of case study	Application process description (including derivation rules and exchange rules)
Stage 3: Create event description	Application information model, Application process description, Textual description of case study	Application event description (event rules)
Stage 4: Create BPMN 'ad hoc' process model.	Application process description,	'ad-hoc' process model that includes data-objects (as in figure 3)
Stage 5: Create application Business process Diagram	'ad-hoc' business process model, Application event description Textual description of case study, Consistent set of assumptions.	BPMN model that excludes data objects (as in figure 5)

6 Conclusion

In this paper we have demonstrated how BPMN can serve the requirement of modeling the (fact-oriented) features in the process- and event perspectives, as expressed in derivation rules, exchange rules and event rules, and the requirement to enable us to record the formal properties of those perspectives that can be used to implement the functionality in software systems. An important role is played by the 'data-object' construct because it contains (instances of) fact types. Another important modeling feature in BPMN that we have applied are the miscellaneous non-graphical attributes that are defined for the core BPMN modeling elements. These non-graphical attributes allow us to capture completely and precisely the remaining formal properties of the conceptual models in the process- and event perspectives of a subject-area. It is recommended to start such an analysis by analyzing 'data use cases' [21, p.7], thereby abstracting them into an application information grammar. Subsequently, a fact-oriented process and event description can be generated. In the fourth stage, the fact-oriented conceptual models can guide us in this process by mapping the said models onto an 'ad-hoc' BPMN process model. In stage five we can analyze the (description of an) application area to create a a 'bird's eye' BPMN model, in which the associated data-objects are left out, to avoid 'clutter' [14, p.93].

References

1. Halpin, T.: Information Modeling and Relational Databases; from conceptual analysis to logical design. Morgan Kaufmann, San Francisco (2001)
2. Verheijen, G., van Bekkum, J.: NIAM: An Information Analysis method. In: IFIP TC-8 CRIS-I conference. North-Holland, Amsterdam (1982)

3. Lemmens, I., Nijssen, M., Nijssen, G.: A NIAM 2007 conceptual analysis of the ISO and OMG MOF four layer metadata architectures. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 613–623. Springer, Heidelberg (2007)
4. Morgan, T.: Some features of state machines in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1211–1220. Springer, Heidelberg (2006)
5. Balsters, H., et al.: Modeling dynamic rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1201–1210. Springer, Heidelberg (2006)
6. Nijssen, G.: Kenniskunde 1A. PNA Publishing Heerlen (2001)
7. Bollen, P.: Conceptual process configurations in enterprise knowledge management systems. In: Applied computing 2006. ACM, Dijon (2006)
8. Morgan, T.: Business Process Modeling and ORM. In: ORM 2007 workshop presentation slides (2007)
9. Balsters, H., Halpin, T.: Formal Semantics of Dynamic Rules in ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 699–708. Springer, Heidelberg (2008)
10. Bollen, P.: Fact-Oriented Business Rule Modeling in the Event Perspective. In: CAISE-forum 2007 (2007)
11. zur Muehlen, M., Indulska, M., Kamp, G.: Business Process and Business Rule Modeling: A Representational Analysisin. In: VORTE 2007 Workshop (2007)
12. Bollen, P.: The orchestration of Fact-Orientation and SBVR. In: 14th international conference, EMMSAD 2009, Amsterdam, the Netherlands. Springer, Heidelberg (2009)
13. Bollen, P.: SBVR: a fact oriented OMG standard. In: OTM workshops 2008, Monterry, Mexico. Springer, Heidelberg (2008)
14. OMG, Business process modelling notation (BPMN) OMG available specification v 1.1. 2008, OMG (2008)
15. White, S.A.: Introduction to BPMN. On demand business Volume (2006)
16. Bollen, P.: Fact-oriented modeling in the data-, process- and event perspectives. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part I. LNCS, vol. 4805, pp. 591–602. Springer, Heidelberg (2007)
17. Bollen, P.: Fact-Oriented Business Service Modeling. In: 12th international workshop on Exploring Modeling Methods in Systems Analysis and Design (EMSSAD 2007), Trontheim, Norway (2007)
18. Patient and Doctor's office 1.5, in White, S. (2006)
19. Nijssen, G.: Semantics for Business: a process to specify the most important business rule in SVBR. Business Rules Journal 8(12) (2007)
20. Bollen, P.: Extending the ORM conceptual schema language and design procedure with modeling constructs for capturing the domain ontology. In: Halpin, T., Krogstie, J., Proper, E. (eds.) Innovations in Information Systems Modeling; Methods and best Practices, pp. 53–67. Information Science Reference, Hershey (2009)
21. Halpin, T., Morgan, T.: Information Modeling and Relational Databases; from conceptual analysis to logical design, 2nd edn. Morgan-Kaufman, San-Francisco (2008)
22. Bollen, P.: A process description language and method for organizational processes in Universal Informatics. In: Second NIAM- ISDM working conference, Albuquerque, New Mexico, USA (1994)
23. Bollen, P.: A Conceptual Modeling Language for the Adequate Design of Business Processes. In: BPMDS 2007, Trontheim, Norway (2007)