Towards an Integrated BPM Schema: Control Flow Heterogeneity of PNML and BPEL4WS

Jan Mendling[1], Cristian Pérez de Laborda[2], and Uwe Zdun[1]

[1] Dept. of Information Systems and New Media, WU Vienna, Austria
jan.mendling@wu-wien.ac.at, uwe.zdun@wu-wien.ac.at
[2] Dept. of Computer Science, Databases and Information Systems,
University of Düsseldorf, Germany
perezdel@cs.uni-duesseldorf.de

Abstract Although there have been standardization efforts for more than ten years, heterogeneity of business process modelling schemas is still a big problem for business process management. This paper discusses the applicability of schema integration methodology in this context and illustrates specific integration problems by discussing the example of BPEL and PNML. Different control flow representations are highlighted as a major challenge in this area. Using classical schema integration and the upward inheritance principle can yield an integrated schema that still includes redundant behavioral concepts. We conclude that future research has to identify extensions to the schema integration process in order to capture such specifics of BPM schemas.

1 Introduction

Heterogeneity of Business Process Modelling (BPM) schemas is a notorious problem for business process management. Although standardization has been discussed for more than ten years, the lack of a commonly accepted interchange format is still the main encumbrance to business process management (see e.g. [1]). A commonly accepted interchange format is needed to move business process models between tools and applications of different vendors. Furthermore, such an interchange format implies the availability of a business process modelling schema that defines the interchange format.

The problem of a missing de facto standard for BPM is addressed by various standardization efforts. Currently, there are at least 15 specifications available or in progress of development, for an overview see e.g. [2]. Different BPM schemas are proposed by the Object Management Group (OMG), the Business Process Management Initiative (BPMI), the Workflow Management Coalition (WfMC), the Organisation for the Advancement of Structured Information Standards (OASIS), the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), the World Wide Web Consortium (W3C), and academic initiatives – some of them addressing only partial aspects of BPM. Recently, various new specifications for Web Service based BPM and Web Service composition including respective XML schemas have been proposed. At least in the short run, they contribute to a further increase of heterogeneity of schemas for business process modelling.

The variety of these standardization efforts raises several questions. Firstly, the interrelation of these various schemas is too little understood. The mere number of specifications does not allow any conclusion on the heterogeneity of concepts included. Research on the comparison of BPM standards like e.g. [3] gives some evidence, but with a narrow focus. Secondly, there is doubt whether standardization processes are a suitable means to come up with an integrated schema for BPM. Weaknesses of BPM standardization have been mentioned in several publications. [4] discusses the diverging strategies of different stakeholders in standardization highlighting the bargaining character of such processes. This holds also for the upcoming Business Process Definition Metamodel standard of OMG. It is required to be UML2-compliant and it will most probably be inspired by IBM products in its major parts. Furthermore, the case of XPDL suggests that standardization may rather lead to specification of the minimal consensual set of concepts than to a consolidation of concepts [5]. In the case of BPEL4WS (or short BPEL) [6] the control flow concepts of XLANG and WSFL were just put together, but semantic redundancies were not eliminated. Accordingly, there is a choice in BPEL between a block structured and a graph structured specification of control flow [7]. In contrast to that, a schema is needed that reflects the superset of concepts available in various BPM specifications and that is free of redundant concepts.

This paper discusses in how far schema integration offers a suitable methodology for deriving an integrated BPM schema from various input schemas. In Section 2 we present the basic ideas of schema integration. In Section 3 we focus on PNML and BPEL as two examples of heterogeneous BPM schemas. We illustrate how schema integration could be used to integrate the schemas and in how far the result is satisfactory. In Section 4 we give an overview on related research in the context of both schema integration and heterogeneity of BPM schemas. Section 5 concludes the paper and gives an outlook on future research.

2 Schema Integration

Schema integration refers to the construction of a global schema from a set of local schemas. In general, the local schemas are heterogeneous, i.e. semantically related concepts are captured by different local schemas in a different way, e.g. using different names or different structure (cf. e.g. [8]). The global schema is expected to be *complete* in capturing all concepts of the local schemas, *minimal* by including semantically related concepts only once, and still *understandable* [9]. Discovering semantic relationships like equivalence, subsumption, intersection, disjointness, and incompatibility between concepts of local schemas plays a central role for schema integration. On the other hand, design criteria like simplicity, completeness, generality, unambiguity, and extensibility have been identified as important for standardization of schemas (see e.g. [2]). Apart from extensibility these criteria match the schema integration criteria of completeness, minimality, and understandability as reported in [9]. As we discussed in Section 1 these criteria are not always met by BPM standards: e.g. XPDL is not complete and BPEL is not minimal. Therefore, schema integration could be a promising methodology to grant completeness and minimality of an integrated BPM schema that could also serve as a candidate standard.

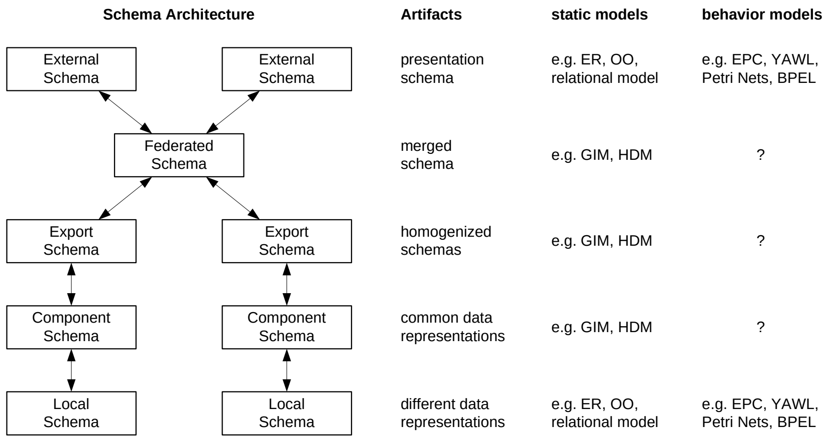

Figure 1. Schema architecture according to [10] and schema integration.

Figure 1 shows a schema architecture as defined in [10]. Typical schema integration problems can be related to this architecture. They are addressed by dedicated transformations from local schemas (bottom) via intermediate steps to external schemas (top). The first step takes the local schemas (that may be represented in heterogeneous data models) as input and transforms them to a common data model representation. This common data model can be low-level or high-level. Low-level models include e.g. the generic integration model (GIM) [11] or the hypergraph data model (HDM) [12]. For a discussion on common data models see e.g. [11]. In the second step, transformations to the export schemas homogenize the component schemas by resolving schema conflicts. The categories of potential conflicts depend on the common data model; a typical example is a concept that is represented by an attribute in schema A and by an object in schema B. In the third step, the export schemas are subject to a merge operation that builds on semantic relationships between concepts of the different schemas. This yields the federated schema which is still expressed in the common data model. In a fourth steps, this federated schema is transformed to external schemas in data models that the user can easily understand. The question is whether schema integration as proposed for static concepts can be applied for deriving an integrated BPM schema. In [11] the straight-forward application of schema integration of static aspects to behavioral aspects is doubted and further research in this area is encouraged. In the following section we discuss problems that may arise when schema integration of static aspects is applied to BPM schemas like e.g. PNML and BPEL.

3 Schema Integration of PNML and BPEL

We will illustrate specific integration problems of BPM schemas by using Petri Net Markup Language (PNML) and the Business Process Execution Language for Web Services (BPEL) as an example. Both define an XML schema. For integration we rely on the definition of intentional relationships like proposed in e.g. [13] and the upward inheritance principle [14].

PNML was designed in order to facilitate the interchange of Petri net models between heterogeneous Petri net analysis tools [15]. PNML includes the standard Petri net elements, i.e. places, transitions, and arcs between them. Furthermore, all these elements can have so-called labels. A label captures the specifics of a certain Petri net type. So-called Petri net type definitions specify the set of allowed labels to define a particular type of Petri net. This extensibility mechanism offers the flexibility to exchange arbitrary Petri net types with PNML. For further details on PNML refer to [15]. *BPEL* is an executable language for the specify Web Service composition. That means BPEL builds on a set of elementary Web Services to define a more complex process that is also accessible as a Web Service. BPEL offers several concepts including *variables* to store workflow data and messages that are exchanged with Web Services. *PartnerLinks* represent a bilateral message exchange between two parties. They are relevant for such *basic activities* that involve Web Service requests. These include the `invoke`, the `receive`, and the `reply` activity. Further basic activities include `wait`, `terminate`, and `assign` to name but a few. Moreover, BPEL offers *structured activities* for the definition of control flow, e.g. to specify sequencing (using `sequence`), concurrency of activities (using `flow`), or alternative branches (e.g. via `switch`). These structured activities can be nested. Finally, there are different handlers in order to respond to the occurrence of a fault, an event, or if a compensation has been triggered. In this paper we concentrate on control flow specification of BPEL, i.e., basic and structured activities. For further details on BPEL see [6].

The example of Figure 2 illustrates a major problem when integrating heterogeneous BPM schemas, i.e. heterogeneous representation of behavioral aspects (also referred to as control flow). There are different formalisms available to represent control flow (see e.g. [16]). Some BPM formalisms are quite different from a syntactical perspective, although they represent similar semantics. Figure 2 gives an example of an AND split with one flow of control branching into two parallel threads of execution. The first grey column provides the XML code for this process in Petri Net Markup Language (PNML), which uses a graph-based representation with places and transitions as special nodes linked via control flow arcs. The second grey column shows the AND split represented in BPEL using a block-structured syntax. The so-called `flow` structured activity is used to specify parallel execution of all its child activities.

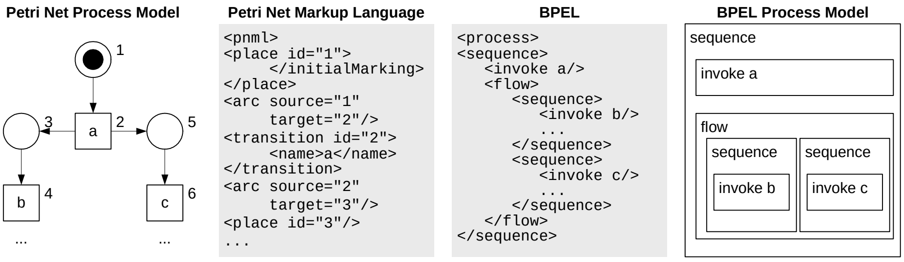

Figure 2. A sample process model with its PNML and BPEL representation

Figure 3 illustrates the PNML and BPEL schemas represented as metamodels. Classical schema integration builds on identifying semantic relationships between the intentional domains of schema constructs, i.e. the real world entities identified by the constructs. In [13] equivalence, subsumption, intersection, and disjointness are defined as intentional semantic relationships. Such relationships need to be considered when merging two schemas. On the right hand side of Figure 3 semantic relationships between the schemas are given:

1. The intention of an invoke activity in the BPEL schema can be subsumed to the intention of a transition in the PNML schema.
2. The intention of a PNML object intersects with that of a BPEL activity, because the intention of a structured activity is beyond the intention of a PNML object, and because there is nothing like a PNML place in BPEL.
3. The intention of a PNML PetriNet intersects with that of a BPEL process. In PNML arbitrary cycles of places and transitions are allowed, BPEL offers only structured cycles in terms of a `while` activity. Yet, BPEL offers the OR join that Petri nets cannot express directly.
4. The intention of the BPEL sequence can be subsumed as a special kind of Petri net. There is always a set of places, transitions, and arcs that can capture the behavior of a sequence modelled in BPEL. This holds also for other BPEL structured activities except the `flow`.
5. The intention of a BPEL activity intersects with a Petri net for the same reason as mentioned in 3.

Figure 4 shows a schema that could be constructed according to the intentional semantic relationships given. Relationship 1 would result in an a merged “transition/basic activity” construct that inherits from BPEL activity and PNML node. This relationship implies already relationship 2. The third relationship results in the creation of a “general process” construct following the upward inheritance principle [14]. This construct also generalizes the BPEL structured activity as given in relationship 5. Finally, relationship 4 motivates the introduction of the structured activity construct, too.

The integrated schema in Figure 4 has still some deficiencies. First, further *simplifications* are possible. Some structured activities can be mapped to Petri nets; e.g. a BPEL sequence can always be expressed as a Petri net. Therefore,

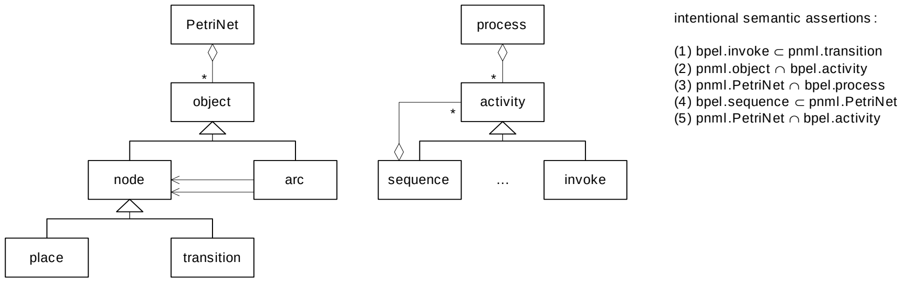

Figure 3. Metamodel of PNML and BPEL and intentional semantic assertions

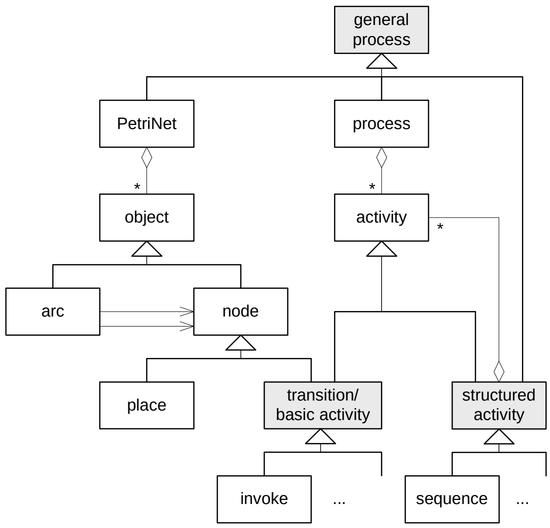

Figure 4. Integrated schema for PNML and BPEL

the sequence is somehow redundant in the integrated schema. The problem is that this kind of redundancy cannot be expressed in terms of a binary intentional relationship, because a BPEL sequence has to be mapped to several nodes and arcs in a Petri net. In order to eliminate this kind of redundancy, both Petri nets and BPEL processes could be mapped to a language with more expressive modelling primitives like e.g. YAWL [17]. Another option could be a mapping to a more basic representation like state charts. Although this representational heterogeneity of behavioral aspects is typical for BPM languages, it seems that it is not inherent to behavior modelling only. Think of two car component schemas: one might use an unordered list of arcs and nodes (analogue to Petri nets) to model subcomponent relationships. The other uses a block-oriented representation by nesting components (analogue to BPEL). The latter allows to model a tree while the first accepts also general graphs. Intentional semantic relationships about mappings between these two different representations could help to eliminate redundant behavioral concepts in the integrated schema. Secondly, as intentional semantic relationships have been used, there is a problem with the *extensional description* of the constructs. A transition in PNML is described by quite different attributes than a basic activity in BPEL. Accordingly, instances stored with the integrated schema would include several NULL attributes, e.g. in the transition/basic activity construct. These problems suggest that a straightforward application of schema integration for static aspects does not yield the desired results.

4 Related Research

As this paper combines BPM research and schema integration, we have divided the discussion of related work into two subsections. The first subsection regards publications dealing with schema integration of XML schemas and of behavioral aspects, whereas the second subsection summarizes research on heterogeneity of BPM languages.

4.1 Schema Integration

Numerous approaches dealing with the integration of heterogenous database schemas have been published so far. We refer to [18] for a detailed overview of different strategies in the context of (semi-)automatic schema matching. A taxonomy of potential conflicts between schema components is given in e.g. [19]. That work builds on an abstract approach regarding real-world objects instead of schema components of relational databases, like it is done in [8]. On the role of identifiers in the integration process refer to e.g. [20,21]. Different data models for schema integration have been proposed, e.g. GIM [11] or HDM [12]. A good overview of research on schema integration in general can be found in [11].

Most BPM schemas are defined as XML schemas. Several approaches deal with the specifics of this schema type. In [22] Behrens addresses the problem of integrating different XML schemas. He proves, that although there is always a new DTD for the intersection of two DTDs, this is not true for their union. Yang et al. introduce in [23] a further XML schema integration concept using a mediator model. Contrary to [22] they transform the XML schema into a semantic rich representation for capturing the implicit semantics stored in an XML schema. Work in the context of the AutoMed project highlights graph restructuring as a promising technique for XML schema integration [24].

The integration of behavioral aspects has received less attention in comparison to integration of static data models. Preuner et al. [25] present an integration strategy for business process models given as a Petri net derivative called object/behavior diagrams (OBD). Yet, heterogeneity of business process modelling schemas is not discussed in this context. Therefore, it is not clear in how far OBD represents a suitable common data model for integration of behavioral aspects. Integration is often related to some notion of inheritance. In [26] four types of inheritance relationships are defined for Petri nets. This work seems to be motivated by model checking as it does not discuss integration aspects.

Yet, there is doubt whether schema integration is suitable as a methodology for standardization of schemas. In [27] schema integration as a bottom-up methodology is contrasted with top-down domain modelling. Schema integration is said to produce schemas that are too much influenced by the local schemas and therefore rather difficult to understand, while domain modelling yields much clearer schemas. The question in this context is how can aspects of domain modelling be included in the integration process to come up with an integrated BPM schema that is clear and straight-forward to understand for a domain expert.

4.2 Heterogeneity of BPM Schemas

In the context of heterogeneous BPM schemas, a lot of research is dedicated to semi-formal comparisons of BPM schemas. Examples include comparisons of BPEL and BPML [3]; DAML-S (predecessor of OWL-S) and BPEL [28]; and XPDL, BPEL, and BPML [29]. Furthermore, a list of 13 high-level concepts of BPM languages has been reported in [2]. This list represents the superset of metamodel concepts extracted from 15 currently available XML-based specifications for business process modelling. Another approach is taken by [30] who identify workflow patterns for control flow semantics. These patterns have been applied as a framework for comparing various BPM languages. Furthermore, that research inspired the specification of a new workflow language called YAWL that is able to capture all pattern (excluding implicit termination). Beyond that, there has been some work on transformations between BPM standards. In [31] a transformation from UML to BPEL4WS is given. Moreover, the BPMN specification includes also a mapping to BPEL4WS. Yet, as both these transformations are one way, it is not clear whether a back transformation is feasible. A general framework for the integration of various BPM schemas is missing.

5 Conclusions and Future Work

In this paper we outlined integration problems in the context of heterogeneous BPM schemas. The example of BPEL and PNML was given to highlight different control flow representations as a major challenge in this area. Using classical schema integration and the upward inheritance principle would yield an integrated schema that could still include redundant behavioral concepts. Basically, further research is needed in at least two areas. Firstly, the specifics of behavioral aspects have to be clearly identified in order to either adapt the schema integration process for static aspects as given e.g. in [11], or to come up with an integration process for behavioral aspects building on a specific common data model. Secondly, the role of such an integration methodology for standardization processes needs to be analyzed. Additional engineering steps might be required in order to further simplify the integrated schemas which represents the output of the integration process.

References

1. Delphi Group: BPM 2003 – Market Milestone Report. White Paper (2003)
2. Mendling, J., Nüttgens, M., Neumann, G.: A Comparison of XML Interchange Formats for Business Process Modelling. In: Proceedings of EMISA 2004 - Information Systems in E-Business and E-Government. LNI (2004)
3. Mendling, J., Müller, M.: A Comparison of BPEL4WS and BPML. In Tolksdorf, R., Eckstein, R., eds.: Proceedings of Berliner XML-Tage. (2003) 305–316
4. zur Muehlen, M., Nickerson, J.V., Swenson, K.D.: Developing Web Services Choreography Standards - The Case of REST vs. SOAP. Decision Support Systems (2005)
5. van der Aalst, W.M.P.: Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language. QUT Technical report FIT-TR-2003-06, Queensland University of Technology, Brisbane (2003)
6. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems (2003)
7. van der Aalst, W.M.P.: Don't go with the flow: Web services composition standards exposed. IEEE Intelligent Systems **18** (2003) 72–76
8. Kim, W., Seo, J.: Classifying schematic and data heterogeneity in multidatabase systems. IEEE Computer **24** (1991) 12–18
9. Batini, C., Lenzerini, M., Navathe, S.B.: A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys **18** (1986) 323–364
10. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput. Surv. **22** (1990) 183–236
11. Schmitt, I., Saake, G.: A Comprehensive Schema Integration Method Based on the Theory of Formal Concepts. Acta Informatica (2005) to appear.
12. Boyd, M., McBrien, P., Tong, N.: The automed schema integration repository. In Eaglestone, B., North, S., Poulovassilis, A., eds.: Advances in Databases, 19th British National Conference on Databases, BNCOD 19, Sheffield, UK, July 17-19, 2002, Proceedings. Volume 2405 of Lecture Notes in Computer Science., Springer (2002) 42–45
13. Larson, J.A., Navathe, S.B., Elmasri, R.: A theory of attribute equivalence in databases with application to schema integration. IEEE Trans. Software Eng. **15** (1989) 449–463
14. Schrefl, M., Neuhold, E.J.: Object class definition by generalization using upward inheritance. In: Proceedings of the Fourth International Conference on Data Engineering (ICDE), IEEE Computer Society (1988) 4–13
15. Billington, J., Christensen, S., van Hee, K.E., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., Weber, M.: The Petri Net Markup Language: Concepts, Technology, and Tools. In W. M. P. van der Aalst and E. Best, ed.: Applications and Theory of Petri Nets 2003, 24th International Conference, ICATPN 2003, Eindhoven, The Netherlands. Volume 2679 of Lecture Notes in Computer Science. (2003) 483–505
16. Mendling, J., Nüttgens, M.: XML-based Reference Modelling: Foundations of an EPC Markup Language. In J. Becker, ed.: Referenzmodellierung - Proceedings of the 8th GI-Workshop on Reference Modelling, MKWI Essen, Germany. (2004) 51–71

17. van der Aalst, W.M.P., ter Hofstede, A.H.M.: Yawl: yet another workflow language. Information Systems **30** (2005) 245–275
18. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal **10** (2001) 334–350
19. Spaccapietra, S., Parent, C., Dupont, Y.: Model Independent Assertions for Integration of Heterogeneous Schemas. VLDB Journal **1** (1992) 81–126
20. Schmitt, I., Saake, G.: Managing object identity in federated database systems. In Papazoglou, M.P., ed.: OOER'95: Object-Oriented and Entity-Relationship Modelling, 14th International Conference, Gold Coast, Australia, December 12-15, 1995, Proceedings. Volume 1021 of Lecture Notes in Computer Science., Springer (1995) 400–411
21. Pérez de Laborda, C., Conrad, S.: A Semantic Web based Identification Mechanism for Databases. In: 10th International Workshop on Knowledge Representation meets Databases (KRDB 2003), Hamburg, Germany, September 15-16, 2003. Volume 79 of CEUR Workshop Proceedings., Technical University of Aachen (RWTH) (2003) 123–130
22. Behrens, R.: A Grammar Based Model for XML Schema Integration. In: BNCOD 17: Proceedings of the 17th British National Conferenc on Databases, London, UK, Springer-Verlag (2000) 172–190
23. Song, I.Y., Liddle, S.W., Ling, T.W., Scheuermann, P., eds.: Resolving Structural Conflicts in the Integration of XML Schemas: A Semantic Approach. In Song, I.Y., Liddle, S.W., Ling, T.W., Scheuermann, P., eds.: Conceptual Modeling - ER 2003, 22nd International Conference on Conceptual Modeling, Chicago, IL, USA, October 13-16, 2003, Proceedings. Volume 2813 of Lecture Notes in Computer Science., Springer (2003)
24. Zamboulis, L., Poulovassilis, A.: Using AutoMed for XML Data Transformation and Integration. In Bellahsne, Z., McBrien, P., eds.: DIWeb Workshop - CAiSE'04 Workshop Proceedings. (2004) 58–69
25. Preuner, G., Conrad, S., Schrefl, M.: View integration of behavior in object-oriented databases. Data Knowl. Eng. **36** (2001) 153–183
26. van der Aalst, W.M.P.: Inheritance of business processes: A journey visiting four notorious problems. In Ehrig, H., Reisig, W., Rozenberg, G., Weber, H., eds.: Petri Net Technology for Communication-Based Systems - Advances in Petri Nets. Volume 2472 of Lecture Notes in Computer Science., Springer (2003) 383–408
27. Hasselbring, W.: The role of standards for interoperating information systems. In Jakobs, K., ed.: Information Technology Standards and Standardization: A Global Perspective. Idea Group Publishing, Hershey, PA (2000) 116–130
28. McIlraith, S., Mandell, D.: Comparison of DAML-S and BPEL4WS. x, Stanford University, http://www.ksl.stanford.edu/projects/DAML/Webservices/DAMLS-BPEL.html (2002)
29. Shapiro, R.: A Comparison of XPDL, BPML and BPEL4WS. Draft version 1.4, Cape Visions, http://xml.coverpages.org/Shapiro-XPDL.pdf (2002)
30. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases **14** (2003) 5–51
31. Gardner, T.: UML Modelling of Automated Business Processes with a Mapping to BPEL4WS. In: Proceedings of the First European Workshop on Object Orientation and Web Services at ECOOP 2003. (2003)