ODD-BP - an Ontology- and Data-Driven Business Process Model

Eric Rietzke[1], Ralph Bergmann[2] and Norbert Kuhn[1]

[1] University of Applied Sciences Trier, Germany
[2] University of Trier, Germany

Abstract. Many researchers have addressed the demands of knowledge intensive processes and often propose a data-oriented workflow approach. Others use ontologies in the periphery of workflow management systems to achieve different kind of contributions, while just a few research utilize the ontology for a semantic process definition.
This paper introduces a new approach which combines both perspectives to define an ontology- and data-driven business process model. The data-driven process characteristic is formed by a metamodel, placed in the base-ontology as the core of the conceptualization. These core concepts are expanded by more specific concepts to build a domain oriented framework for the enterprise and process knowledge. Aligned with an example, we will explain how process definitions are represented in the knowledge store and examine the gradual transition of an executable process instance. As a result, the ODD-BP approach takes advantage of a declarative data-oriented process model regarding flexibility, while the semantic process definition reduces ambiguity and builds the foundation to drive the process execution through inference mechanisms.

1 Introduction

Compared with the established and well known BPMN approach, the new approach introduced with this paper appears to be odd, but despite this coincidence, ODD-BP stands for ontology- and data-driven business process approach. The motivation arises from the research about knowledge intensive processes (KiPs) with its data-oriented character [5] and the general search for AI-support within business processes. Established approaches often place data as a third class citizen into the process definition, sometimes expressed as an afterthought [4] within a classical control-flow oriented model. KiPs are knowledge- and data-centric and require flexibility at design and run-time [3], especially regarding their major resource, the knowledge workers. They should be supported by offering opportunities rather than restrictions, an accompanying system should deliver choices and recommendations and access to relevant information to accomplish a contribution during the process execution [3, 5].

Artificial intelligence (AI) in all its facets and across the different kind of technologies are discussed in a wide range of use-cases as well as it is in the focus of research in the BPM-context. No matter if an AI contribution is delivered by the workflow system itself or by an external agent, when it comes to a division of labour between human- and cyber actors, the process must be described without ambiguity and understandable for all process participants. An ontology is the perfect tool to achieve this requirement and can be used for a semantic process definition according to Fellmann [24].

The combination of both leads to an ontology and data driven ODD-BP approach as it is described in this paper. The data-driven character is formed by a metamodel defined in the *base-ontology* as the core of the conceptualization. Aligned with our research project SEMANAS[3], we examine the demands of knowledge intensive process in the domain of agricultural grant applications and expand the base-ontology by a *domain-ontology* focused on this specific use-case.

Section 2 is referring the foundations of our work, while section 3 introduces the metamodel for our new approach. The application of the metamodel is described in section 4, while additional transition rules expand the conceptualization to ensure a valid execution of process instances. Section 5 gives a short insight into our current development of a process design-tool. With the conclusion in section 6, we give an outlook about the possibilities and advantages of the ODD-BP approach and our future work.

2 Foundations

Business Process Model and Notation (BPMN) is currently the de facto standard for designing and describing business processes world-wide. In the center of this approach reigns a control-flow coordination of process steps (activities). A less restrictive, but still activity-centric perspective is provided by constraint-based approaches [19], which allow flexibility in a scalable manner. Alternatively, there are several approaches with the intention to gain flexibility based on the control-flow principles [23, 16]. Despite of the consideration of data-flow in such processes, the data is just integrated in a kind of an afterthought [4, 11]. Opposed to this, knowledge-intensive processes (KiPs) are usually barely structured and their execution is driven by user decisions and business data. Previous research has shown [25, 2, 17] that an activity-centric perspective is not sufficient to support knowledge-intensive processes.

With view to these insights, several new approaches were brought up during the last decade, putting the data into the center. The case handling paradigm [25] elevated the result of a process (case), reflected by its data objects; activities do not longer drive the process but serve the outcome. For more complex scenarios with the need of abstraction capabilities, object-awareness approaches refined the case handling concept[26]. With business artifacts [1, 2], CorePro [17], and PHILharmonicFlows [15] there are even more approaches to mention, which

[3] SEMANAS is funded by the Federal Ministry of Education and Research (BMBF), grant no. 13FH013IX6, duration: 2017-2021

underlines the importance of data-centric approaches for knowledge intensive processes.

Beside the examinations of different workflow models and principles, the possibilities of semantic information systems in the domain of BPM are matter of current research as well. S-BPM introduced a subject-oriented modeling scheme, where sentences with subject, predicate, object are used to describe general interactions between process actors [8]. This profound methodology addresses the communication aspects between process actors, but offers no specific strategies regarding the demands of KiPs. The initiative of WSMO (Web Service Modeling Ontology) uses ontologies to formalize the interoperability of web services [7] and thus targets a service orchestration purpose. Some work considers how a formal semantic can be utilized for process validation [20] or optimization [9] purposes. Opposed to this, Thomas and Fellmann [24] introduced an ontology based representation of business processes in which process elements are assigned to ontology classes to define a control-flow oriented metamodel. Further research also uses semantic process modeling to define control-flow oriented approaches like [12, 13]. In general, the research about semantic process definitions is motivated by the reduction of ambiguity [6, 24] and the opportunity of inferencing new knowledge [14] on base of the existing process knowledge.

The research about semantic formalizations within the field of business processes defines the foundation for our work, but to the best of our knowledge, no research addresses a semantic process modeling principle for a data-oriented workflow approach to support the definition and execution of knowledge intensive processes. The general idea for this approach was already introduced through our precedent work [22, 21] and will be carried forward with this paper.

3 Metamodel

According to a wide range of publications [10, 18], a metamodel defines "the frames, rules, constraints, models and theories applicable and useful for modeling a predefined class of problems." Knowledge intensive processes (KiPs) and their specific demands [5] can be considered as such a predefined class of problems. We define the ODD-BP model, a workflow metamodel aligned to the requirements of KiPs, utilizing the underlying ontology to provide semantic and data-oriented process definitions.

3.1 Mapping between Knowledge Store- and WfMS-Structure

Any workflow management system follows a metamodel, usually implementing one of the workflow-approaches mentioned in section 2. From a most fundamental perspective, all WF-approaches have in common, that they use a set of activities to achieve a specific goal. The *metamodel* defines the kind of process elements and their possible interactions and based on this model, a *process definition* (PD) is specified for each kind of process goal, acting as a blueprint for *process instances* (PI) which can be executed to achieve a specific goal of a certain kind.

All established workflow approaches follow this general WfMS-structure built on a metamodel, process definitions and process instances (Fig. 1).

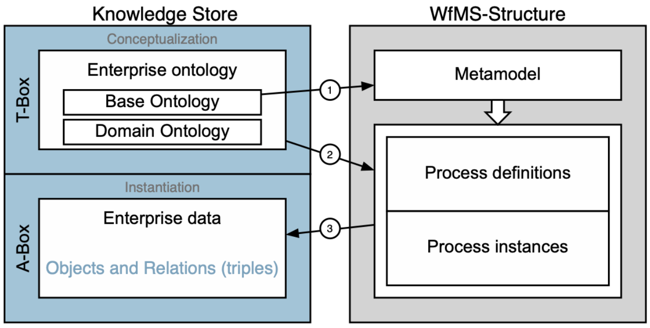

Fig. 1. Modules of the knowledge store and the relations to the WfMS-structure

The knowledge store (KS) is the combination of the ontology (T-Box) as the conceptual fundament and the triple store (A-Box) as the data storage. The *base ontology* defines the metamodel ① of the ODD-BP approach, it is domain independent and will be introduced in detail in 3.2. The domain ontology expands the conceptualization by defining general valid concepts and relations of a domain. Further ontologies can expand the conceptual knowledge, like a document ontology, and form together the enterprise ontology, the fundament ② of all process definitions and process instances. A process definition or a process instance is defined by a set of linked individuals according the concepts and relations of the metamodel. Such a set represents a process graph and is stored as triples ③ in the A-Box.

3.2 Base-Ontology

The base-ontology defines all concepts and relations to build the metamodel of the ODD-BP approach. The most fundamental artifacts of a process are *Tasks*, *Dataobjects*, *Documents* and *Actors*. Individuals of these concepts can be connected with an individual of the class *Process* through a *contains* or *involves* relation, as it is shown in Figure 2.

The mentioned artifacts are usually represented in one or another way in all workflow approaches. The specific character of a metamodel is manifested by the kind of relations between these artifacts and in this case the data-oriented character of the ODD-BP approach is created by the relations between *Tasks* at the one side and *Documents*, *Dataobjects* and *Attributes* on the other side.

According to this metamodel, a *Document* can be *demanded_by* a *Task* as input or a *Task* can *produce* a *Document* as the outcome of its execution. Analog to this, a *Dataobject* or an *Attribute* can be *required_by* a *Task* as input or a

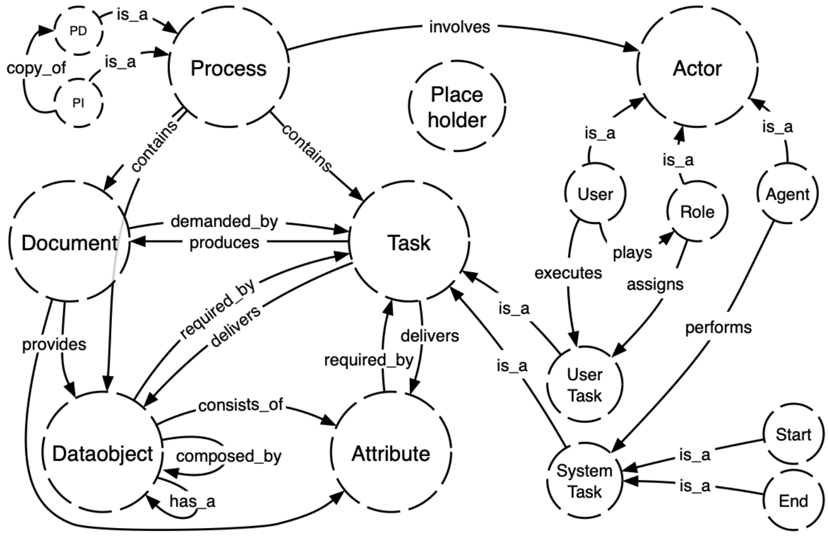

Fig. 2. Base-Ontology

Task can deliver such an element as output. The deeper meaning of *Dataobject* and *Attribute* will be explained more in detail in the following, but the general importance of data for a process execution is obvious.

Unlike these direct relations between the named concepts, the possible relations between *Actors* and *Tasks* are not defined in a direct manner, but through specialized concepts. This allows a precise definition of different kind of *Actors* and *Tasks* and their individual relations. Corresponding to this, a cyber actor (labeled as *Agent*) can *perform* a *System Task*. A *Role* can just be *assigned* to a *User-Task*, while a *User* can *execute* such a *User-Task*, if the *User* is allowed to *play* the corresponding *Role*.

All in all the metamodel defines, how a process can be designed and executed and these general rules are the same for process definitions *PD* and process instances *PI*, which are defined as a specialization of the concept *Process*. A process is modeled and described by individuals of the introduced concepts and by links between these individuals according the relations of the base ontology.

3.3 Data-Orientation and Data-Driven

The general data-orientation of the metamodel can be seen by the manyfold relations between *Task* and the data-carrying elements *Dataobjects*, *Attributes* and *Documents*, in certain ways similar to artifact centric approaches [1, 2, 4]. However, the ODD-BP approach is not only placing data into a more central position, the data is integrated with the intention to drive a process.

Usually an information system organizes data about the real world with entities and relations. With view to databases, entities are managed as entries into a table, while a knowledge store is managing entities as individuals of a certain class. Such an individual represents an object of the real world, while its object-characteristics are stored as data-properties of the individual. This realization lacks in expressiveness, as the knowledge store can not express dependencies between tasks and data-properties and thus, the data-properties could not be used to drive the process.

This leads us to a conceptualization in which object-characteristics are represented through a separate concept, Attributes. They can be understood as key-value pairs, while an individual of this type is representing a single characteristic of an entity. With view to the example shown in Fig. 3, the individual *Person* is a *dataobject* to represent a specific person in the real world. The *birthdate* and the *adult state* are represented by 2 separate individuals of the base-type *Attribute*, linked with *Person* through a *consist_of* relation.

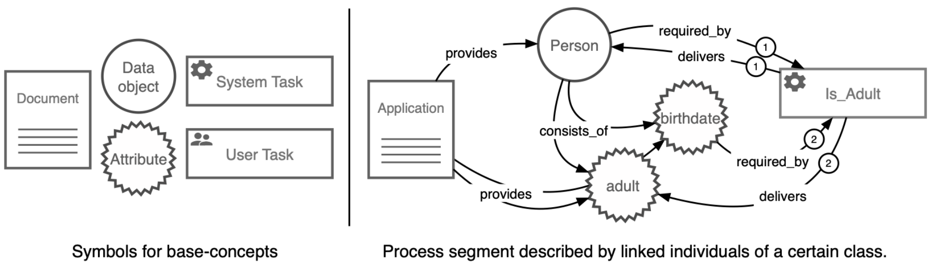

Fig. 3. Process visualization of a data-driven process segment

Assuming that a system task *Is_Adult* can decide autonomously if a Person can be seen as adult or not. The relation between dataobject and task would usually just be expressed at an abstract and more general level ①. Through the additional conceptualization, the relation between the system task *Is_Adult* and the *Attributes* ② can be modeled more in detail. This offers a more specific process definition which allows to deduce the executability of tasks by an inference engine using the linked (input-)attributes with a *required_by* relation. Beside the pure executability, such activities can deliver a more or less important process contribution to achieve a process goal which can be deduced using the linked (output-)attributes with a *delivers* relation.

Beside the entities itself, an information system also allows to express relations between entities. The base-ontology (Fig. 2) defines two different base-types to express such relations. The *has_a* relation is intended to define any kind of global valid relations between entities. Within a further conceptualization through a domain ontology, specializations of the *has_a* relation can be used to form an enterprise information model, equivalent to an ER-model of a database.

The *composed_by* relation is the prototype of a more process dependent combination of entities, not valid from a general perspective, but valid and required within a process context. It can be used to combine different kind of dataobjects for a process definition and process execution.

3.4 Semantic Integration of Documents

Data can come from different sources like an information system, from a user or cyber actor (through a process or system task) and by documents, which usually means data-exchange with external systems or users. From this perspective, a *document* can *provide dataobjects* (Fig. 3) within a process context. According the introduced design to represent entities, a *document* can also *provide attributes* of *dataobjects*. These *provide* relations are also introduced by the base ontology (Fig. 2) and build the foundation for a semantic integration of documents into a process context. This topic is object of our current research and expands the possibilities of the ODD-BP approach even further.

4 Application of the Metamodel

The introduced metamodel defines the vocabulary, which allows a process designer to express a process definition for a certain process goal. The data of process definitions and process instances as well as any data of the information system is part of the knowledge store and is persisted as triples in the triplestore, the A-Box. Since any process-element within a process definition is just used to form a process template, these elements are placeholders and with view to a process instance, such placeholder elements will be replaced by meaningful elements during the process execution. To express this, the base-ontology (Fig. 2) also defines a *placeholder* concept. Any process-element of a PD is represented by an individual assigned to one of the introduced concepts (task, dataobject, attribute, document) and additionally assigned to the *placeholder* class. This serves the inference mechanism to deduce the executability of activities as we mentioned before and which will be addressed in detail by a separate paper.

4.1 Process instance

A process instance is initially nothing else than a copy of a process definition with all its placeholder elements. Along the execution, the process instance performs a gradual transition from a process description defined only by placeholder elements to a final process state, where some or all process elements are meaningful elements. This gradual transition must follow some rules, but since the process descriptions of PD and PI are defined by a set of individuals in the A-Box, the rules can not be expressed using the conceptual layer of the knowledge store. In the following we will introduce these rules which define valid structural changes between a modeled process definition and corresponding executed process instances.

4.2 Multi-instantiation

A PD defines a template to achieve a process goal like an application for a group event. In such a PD, a single individual of the classes dataobject and placeholder is representing the applicants, not knowing how many applicants will finally participate in a PI. To express this aspect of a multi-instanciation, the metamodel must offer the option to define the cardinality for each process element (PE). Since the placeholder PEs will be replaced during the process execution, the right place to persist the possibilities of the cardinality is at the link between the individuals of the process p and the process element pe. For this purpose, the *contains* relation is expanded by an annotation, which allows each link of this type to define the cardinality by a list with 2 numbers (n, m).

- $n \in \mathbb{N}_0$ defines the minimum occurrence of a process element, while $n = 0$ defines an optional occurrence within a PI.
- $m \in \mathbb{N}_0$ defines the maximum occurrence of a process element, while $m = 0$ defines an unlimited occurrence within a PI.
- In the following, the cardinality will be expressed with an additional label at the *contains* relation as: $p \xrightarrow[(n,m)]{contains} pe$

The multi-instanciation only defines the possible occurrence of PEs. It tells nothing about the way, how such PEs interact with each other. This requires a further extension to express the intended transitions between PEs.

4.3 Transition rules

According the metamodel, *Tasks*, *Documents* and *Dataobjects* are connected with a *Process* through the *contains* relation. The interplay between individuals of these three concepts are defined by the five relation types *provides*, *demanded_by*, *produces*, *required_by* and *delivers*. Analog to the ontology, two connected process elements can be seen as a process statement with subject, predicate and object, while subject and object are individuals of the three mentioned concepts and predicate is a link of the five mentioned relations. Such a process statement defines an action (according the predicate) from a subject to the object. Since subject and object can be defined with a different cardinality, the predicate must be extended with an annotation how the multi-instanciation can be processed to perform the transition from placeholder PEs to meaningful PEs. We can differentiate in three transition rules:

- expand: Each subject can result (along the predicate) in one or many objects.
- maintain: Each subject can result (along the predicate) in exactly one object.
- join: All subjects can result (along the predicate) in exactly one object.
- In the following, the transition rules will be expressed with an additional label at the predicate as: $pe \xrightarrow[(transition-rule)]{predicate} pe$

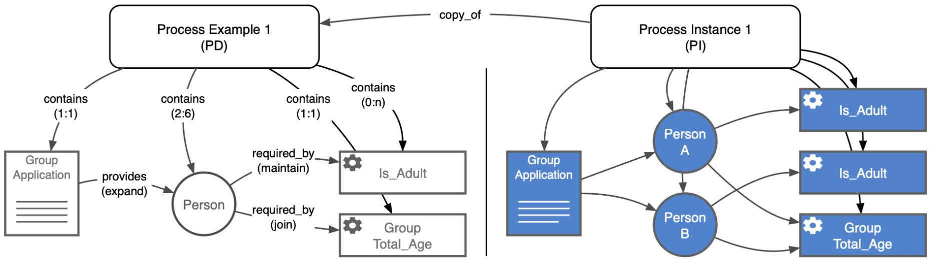

Fig. 4. Multi-instantiation and transition rules

As an example, Figure 4 shows a small process segment of a group application with different cases of multi-instanciation and transition rules. The left side presents the process definition, where the links of the *contains* relation are annotated with the definition for a multi-instanciation, while the other links are annotated with the transition rules. The PD defines, that exactly one (1:1) *Group Application* is allowed, it allows at least 2 and a maximum of 6 *Persons* (2:6), any number of *Is_Adult* tasks and just exactly one task *Group Total_Age*. These four process elements are linked through *provides* and *required_by* relations with the three different transition rules (*expand*, *maintain*, *join*).

The right side presents a corresponding process instance, where all process elements are transformed from placeholder PEs to meaningful PEs, according the introduced rules. The *Group Application* has *provided* two *Persons*, which fulfills the demand of the *contains* relations and follows the *expand* transition rule. It is *required*, that each *Person* must be an adult and the *maintain* transition rule is defining, that for each *Person* an own *Is_Adult* task must be executed. Additionally, the total age of the group is needed, which is why both *Persons* are linked with just one *Group Total_Age* activity according the *join* transition rule.

5 Designing a Process

The metamodel defines the vocabulary to describe a process and through the introduced rules regarding multi-instantiation and transitions, also the semantic dependencies between process elements can be expressed. For a practical use and to design and execute ODD-BP definitions and instances, a modeling tool is required.

As a first step towards a POIS, we started the development of a graphical toolset on a web-based client-server architecture. It aims to support a process designer by utilizing the metamodel to turn the ontological restriction into an easy to use graphical user-interface. Beside this, it also sets the outer ontological restriction into action.

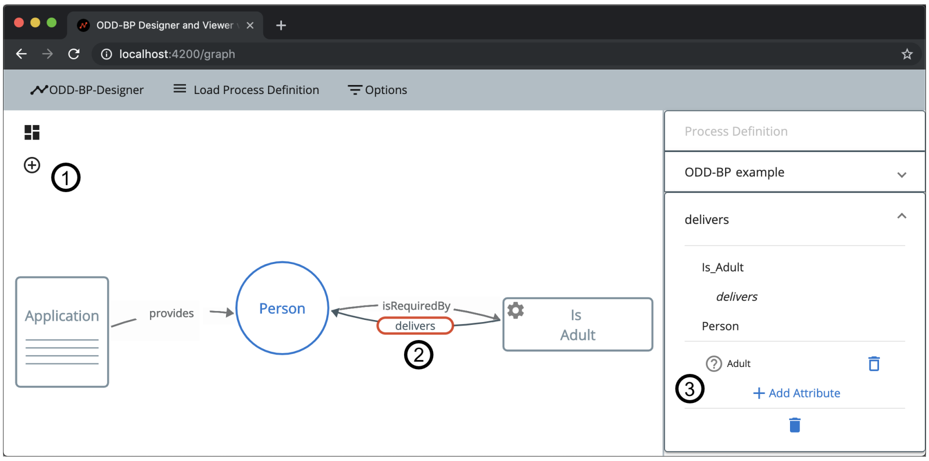

Fig. 5. ODD-BP-Designer

Fig. 5 shows our ODD-BP Designer with the process segment from Fig 3. The small + symbol ① allows to add valid process elements to the current PD according the base- and domain-ontology, actually all elements which can be connected to the PD through a *contains* relation. The editor allows to connect process elements ② like the dataobject *Person* and the system task *Is_Adult*, also utilizing the conceptualization. The *delivers* relation between *Is_Adult* and *Person* can be specified more in detail by connecting the task with specific attributes of the *Person* like the attribute *Adult* ③.

The editor is highly generic since the possibilities to form the process is ascertained through the ontologies. Beside the visual interpretation of the process definition, the tool is taking care for a consistent process model by realizing the outer ontological restrictions. As an example: As soon as the user is deleting the linkage between *Is_Adult* and *Person*, the editor will also remove all links to attributes of the dataobject *Person*. The ontology defines no limitations for this case, which is why the editor has to fill this gap to prevent inconsistency.

6 Conclusion

AI technologies in general can deliver a wide range of contributions within the field of business processes and even semantic information systems can be used in different ways. The introduced ODD-BP approach combines the principles of semantic process definitions [24] with a metamodel which implements a declarative and data-oriented process character. Thus, it reduces ambiguity and supports the division of labour between human and cyber actors and takes advantage of a data-oriented approach according the demands [5, 3] to design and execute knowledge intensive processes. Through the renunciation of control-flow principles and our focus to a descriptive process model, we gather the advantages of none imperative approaches [11, 19], regarding flexibility during the process execution.

One central motivation is to utilize the ontology and the data-oriented metamodel to drive the introduced approach, which leads us to the acronym ODD-BP approach. This requires a precise definition of the interplay between data and activities and we have shown, that the conceptualization of *Attributes* through the metamodel expands the expressiveness of a process definition and defines the base for different kinds of cyber process contributions. Within a separate paper we will show, that a process definition on base of the ODD-BP approach can be used to deduce the executability of activities. Further more we will show, that even the relevance of an activity can be deduced according its contribution to reach predefined process goals and process milestones.

There is a wide range of further possibilities to take advantage of the ODD-BP approach. As such, the explainability of the inference while deducing executable tasks could be utilized to adapt a process instance according an identified problem. The semantic process definition could also be used for an adaptive process visualization as it was already introduced [22, 21] and which is object of our ongoing research. Finally, the practical use of our new approach within different knowledge intensive application scenarios must be examined as well.

References

1. Bhattacharya, K., Gerede, C., Hull, R., Liu, R., Su, J.: Towards formal analysis of artifact-centric business process models. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **4714 LNCS**, 288–304 (2007)
2. Bhattacharya, K., Hull, R., Su, J.: A data-centric design methodology for business processes. Handbook of Research on Business Process Modeling pp. 503–531 (2009)
3. Bromberg, D.: BPM for Knowledge Workers: The Structural Foundations of Decision Intensive Processes (DIPs). BPTrends (January), 1–7 (2007)
4. Cohn, D., Hull, R.: Business artifacts: A data-centric approach to modeling business operations and processes. IEEE Data Eng. Bull **32**(3), 3–9 (2009)
5. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. Journal on Data Semantics **4**(1), 29–57 (2015)
6. Fan, S., Hua, Z., Storey, V.C., Zhao, J.L.: A process ontology based approach to easing semantic ambiguity in business process modeling. Data and Knowledge Engineering **102**, 57–77 (2015). https://doi.org/10.1016/j.datak.2016.01.001, http://dx.doi.org/10.1016/j.datak.2016.01.001
7. Feier, C., Polleres, A., Dumitru, R., Domingue, J., Stollberg, M., Fensel, D.: Towards intelligent web services: the web service modeling ontology (WSMO). Design pp. 179–196 (2005)
8. Fleischmann, A., Stary, C.: Whom to talk to? A stakeholder perspective on business process development. Universal Access in the Information Society **11**(2), 125–150 (2012). https://doi.org/10.1007/s10209-011-0236-x
9. Francescomarino, C.D., Corcoglioniti, F., Dragoni, M., Bertoli, P., Tiella, R., Ghidini, C., Nori, M., Pistore, M.: Semantic-Based Process Analysis. The Semantic Web ISWC 2014 pp. 228–243 (2014)

10. Gaur, A., Richariya, V.: A Layered Approach for Intrusion Detection Using Meta - modeling with Classification Techniques. International Journal of Computer Technology and Electronics Engineering (IJCTEE) **1**(2), 161–168 (2008)
11. Giorgio, B.: A Data-flow Language for Business Process Models. Procedia Technology **16**, 128–137 (2014)
12. Heinrich, B., Bewernik, M.a., Henneberger, M., Krammer, A., Lautenbacher, F., Bewernik, M.: SEMPA - Ein Ansatz des Semantischen Prozessmanagements zur Planung von Prozessmodellen SEMPA Ein Ansatz des Semantischen Prozessmanagements zur Planung von Prozessmodellen **50** (2008)
13. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: A vision towards using semantic web services for business process management. Proceedings - ICEBE 2005: IEEE International Conference on e-Business Engineering (1). https://doi.org/10.1109/ICEBE.2005.110
14. Koukovini, M.N., Papagiannakopoulou, E.I., Lioudakis, G.V., Dellas, N., Kaklamani, D.I., Venieris, I.S.: An ontology-based approach towards comprehensive workflow modelling. Software, IET **8**(2), 73–85 (2014). https://doi.org/10.1049/iet-sen.2013.0037
15. Künzle, V., Reichert, M.: PHILharmonicFlows: Towards a framework for object-aware process management. J. Softw. Maint. Evol. **23**(4), 205–244 (2011)
16. Minor, M., Bergmann, R., Görg, S.: Case-based adaptation of workflows. Information Systems **40**, 142–152 (2014)
17. Müller, D., Reichert, M., Herbst, J.: Data-driven modeling and coordination of large process structures. In: Proc. CoopIS'07. vol. 103, pp. 131–149 (2007)
18. Neto, A.T., Bussamra, F.L.d.S., e Silva, H.A.d.C.: A new metamodel for reinforced panels under compressive loads and its application to the fuselage conception. Latin American Journal of Solids and Structures **11**(2), 223–244 (2014). https://doi.org/10.1590/S1679-78252014000200005
19. Pesic, M., Schonenberg, H., Van Der Aalst, W.M.P.: DECLARE: Full support for loosely-structured processes. Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC pp. 287–298 (2007)
20. Pham, T.A., Nguyen, T.H.H., Le Thanh, N.: Ontology-based workflow validation. In: Proceedings - 2015 IEEE RIVF International Conference on Computing and Communication Technologies: Research, Innovation, and Vision for Future, IEEE RIVF 2015. pp. 41–46 (2015). https://doi.org/10.1109/RIVF.2015.7049872
21. Rietzke, E., Bergmann, R., Kuhn, N.: Adaptive business process visualization for a data and constraint-based workflow approach. In: CEUR Workshop Proceedings: LWDA. vol. 1917 (2017)
22. Rietzke, E., Bergmann, R., Kuhn, N.: Semantically-Oriented Business Process Visualization for a Data and Constraint-Based Workflow Approach. In: BPM 2017 International Workshops. pp. 142–150. Springer, Cham (sep 2018), http://link.springer.com/10.1007/978-3-319-74030-0_10
23. Sauer, T., Minor, M., Bergmann, R.: Inverse Workflows for Supporting Agile Business Process Management. Proc. 6th Conf. Knowl. Manag. **182**, 194–203 (2011)
24. Thomas, O., Fellmann M.A., M.: Semantic Process Modeling Design and Implementation of an Ontology-based Representation of Business Processes. Business & Information Systems Engineering **1**(6), 438–451 (2009)
25. Van Der Aalst, W.M.P., Weske, M., Grünbauer, D., der Aalst, W.M.P., Weske, M., Grünbauer, D., Van Der Aalst, W.M.P., Weske, M.: Case handling: A new paradigm for business process support. Data Knowl. Eng. **53**(2), 129–162 (2005)
26. Weber, B., Reichert, M.: Enabling Flexibility in Process-aware Information Systems (2012)