Ontology-based representation of workflows for transfer learning

Miriam Herold and Mirjam Minor

Wirtschaftsinformatik, Goethe University, Robert-Mayer-Str.10, Frankfurt am Main, Germany,
{herold, minor}@informatik.uni-frankfurt.de

Abstract. This paper examines the feasibility of using ontologies for transfer learning in process-oriented contexts. Transfer learning uses the knowledge learned in a source domain to improve the ability to solve problems in a target domain. Ontologies can help to store the domain knowledge with all appropriate relations between the concepts. This work describes an approach for capturing workflows represented in BPMN language in an ontology. The aim is the transfer of procedural knowledge from a source into a target domain and we study the feasibility of using a process-oriented ontology as a means for the transfer. We illustrate the approach with an ontology of workflows from passenger and baggage handling at the airport. For creating the ontology we transform airport workflows from a proprietary format into BPMN and use the main elements (pool, lane, sequences etc.) for depicting the workflow structure. Then we represent all existing workflows by the ontology. The suggested approach is generic and domain-independent. It allows broad opportunities for transfer of procedural knowledge.

Keywords: transfer learning, process-oriented case-based reasoning, ontology

1 Introduction

Management of business processes is a widespread area in the business context. Recently, many enterprises face new challenges such as digital transformation and, thus, need to adjust their business processes. Digital transformation in the current context means the transformation of key business operations and affects processes and production, as well as management concepts and structure of an organization (Matt et al., 2015). But there are many other purposes and business areas that require flexibility and adaptation of existing business processes (Minor et al., 2014a). To overcome these challenges companies need to adapt the existing workflows according to the changed conditions. *Case-based reasoning* (CBR) may provide a support for this purpose as it is based on the intuition, that similar problems tend to have similar solutions (Richter and Weber, 2016). After an appropriate adaptation the past solution may help in solving the current problem. In all settings where workflows are involved, CBR methods can be extended for process management. *Process-oriented case-based reasoning* (POCBR) systems *'are capable to support the creation and adaptation of workflows by reasoning on cases recording experiential knowledge from previous workflow modelling, execution, or monitoring activities'* (Minor et al., 2014b). A *case* in POCBR is typically a process description or a workflow expressing procedural experiential knowledge (Minor et al., 2016).

In many domains the knowledge of business processes and workflows is not excessive. Then it is usefull to examine if things learned in one well-known domain can be adapted and re-used in another related context (Kudenko, 2014). *Transfer learning* (TL) addresses this question. In the context of CBR, TL uses knowledge in a source domain to improve the ability to learn to solve tasks in a target domain, where the knowledge is sparse (Klenk et al., 2011). This paper is motivated by the idea of transfering the process-oriented knowledge from a familiar domain to another one lacking in expertise. In the area of business processes there still exists little research on transferability of cases. We would like to address this research gap and aim to find a novel approach to achieve the knowledge transfer.

Previous work on the transferability of process-oriented cases (Minor et al., 2016) has been based on manual ontology construction. In this paper we plan to extend this idea and examine to find an automated ontology-based transfer learning approach. It is the first stage of the ongoing project EVER2 [1]. We use ontologies as a knowledge base, as it allows a flexible representation of wide-ranging relations and concepts. Relations are a very important issue in analogical models, which have been widely examined in the context of transfer learning. To be able to capture relations of various complexity, we decided to use ontologies for the representation of procedural knowledge and as a means for transfer. Ontology is defined as an *'explicit specification of a conceptualization'* and includes concepts, relationships, and other distinction relevant for modelling a domain (Gruber, 2009). In our current work, workflows are annotated by concepts of an ontology, which contains transfer rules for the knowledge transfer from the source in the target domain. For the demonstration we used the open-source tool Protégé. The created ontology represents a case-base for workflows. In the past decade the Business

[1] funded by Deutsche Forschungsgemeinschaft (DFG) under the project number MI 1455/2-3

Process Model Notation 2.0 (BPMN) disseminated across enterprises and industries. It is a standard graphical language for the specification of business processes. Our suggested procedure for transformation via ontology is based on workflows modelled in BMPN 2.0.

This paper is structured as follows: In the next section we provide an overview of the related work. In the third section we introduce an approach for ontology-based representation of BPMN-workflows. Then we demonstrate the feasibility of our proposed approach with an example. The paper continues with a draft for the next possible project direction and future research opportunities. In the last section we summarize our work and draw some conclusions.

2 Related Work

Transfer learning has been examined in many different contexts, especially in machine learning (Taylor and Stone, 2009), (Kudenko, 2014) or data mining (Pan and Yang, 2010). But there is still little research in the area of CBR. CBR systems collect problem-solution pairs (cases) in a case-base and are able to learn, retrieve, adapt and use this knowledge for solution of new upcoming problems (Richter and Weber, 2016). The existing adaptation approaches in CBR can perform a kind of TL-strategy. (Klenk et al., 2011) describe in their work 'CBR as a transfer learning method'. Our research is a contribution to process-oriented case-based reasoning and we plan to examine ontologies as a means for transfer of procedural knowledge.

In case-based TL there is a considerable amount of research on using analogical models ((Falkenhainer et al., 1989), (Klenk and Forbus, 2013), (Könik et al., 2009), (Kuhlmann and Stone, 2007), (Ragni and Strube, 2014)). Analogy represents the overlap between the source and the target domain. If the knowledge is stored in a hierarchical structure the overlap can be located on a higher hierarchical level. The workflows in a case-base have to be abstracted according specific abstraction rules in the source domain and refined in the target domain. Generalization and specification can also be used as a means for transfer learning (Müller and Bergmann, 2015). (Müller and Bergmann, 2014) propose a compositional adaptation approach using *workflow streams*. They identify substitutable components of a workflow and replace them by other suitable workflow streams. In our project we plan to extend this research ideas and develop a system for automated abstraction of process-oriented cases. The aim is to find an appropriate abstraction level, where the overlap is sufficient and allows the knowledge transfer between two domains.

Process ontology based approach (POBA) as in (Fan et al., 2016) uses ontology primarily to ease semantic ambiguitiy in modelling of business processes. In the first step of the POBA approach the authors transform an existing non-process domain ontology in a process ontology. In contrast to our approach, which focuses on transfer learning, their goal is different. They capture semantic concepts in an unambiquous manner in order to improve the efficiency and quality in modelling of business processes. (Montani and Leonardi, 2014) developed a framework for supporting of run-time adjustmets and a subsequent analysis of business processes. Their approach allows retrieval of process traces (recorded execution order of tasks) similar to the current process. Our focus is rather on the process models than on the post-mortem analysis of executed workflow traces.

In the literature there are existing BPMN-ontologies that capture all elements of BPMN specification (Natschläger, 2011) and (Rospocher et al., 2014). The main focus in our work is not building an ontology with all possible elements of BPMN 2.0, but primarily in capturing the existing business processes with the *control flow* and the *data flow*, which are characteristic features of a workflow. The control flow determines the order of task execution whereas data flow specifies how data items (information or documents) interact with the tasks. The proposed ontological representation can be continually extended with new elements based on the increasing workflow repository.

In the past there has been done reasonable effort in the research of semantically enhanced business process modelling (Abramowicz et al., 2012), (Thomas and Fellmann, 2009). The two models describe the conception and creation of process-oriented ontologies, partly based on BPMN-workflows. The findings are demonstrated on some examples, but the results as a whole were not publicly available during the initial phase of our project. To be able to implement our future work we decided to create an own ontological representation based on the characteristics of workflows from our repository.

3 Ontology-based representation of BPMN-workflows

The ontology-based representation of BPMN-workflows introduced in this section can be used for different domains. Only the creation of taxonomy structures requires domain expert knowledge. To build the workflow-ontology we follow four steps. Steps 1 and 2 represent terminological knowledge (TBox), while steps 3 and 4 consider assertive knowledge (ABox) (Baader et al., 2004).

Step 1

First we created basic *classes* for:
- *Structural parts* as events, gateways, processes (incl. sub-processes) and tasks
- *Actors*, corresponding to lanes in BPMN
- *Documents*, which could be consumed or created by tasks

Step 2

In the second step we generated *object properties* to capture all existing relations between the elements of a workflow. Object properties can be stored in a taxonomic structure. We organize them in three classes:
- *ActivityPerformingRelations*, the main relation in this class is *'do'* to show, for example, that an actor is responsible for executing a specific task
- *AssignmentRelations* to express that an element belongs to one specific workflow or a document is input/output of a particular task
- *TemporalRelations* to capture the order of tasks, events, gateways etc.

Step 3

Based on the available BPMN-workflows we create *instances* for all actors, documents and structural parts.

Step 4

The last step is the *representation of all properties* of a workflow, using the object properties and instances created in previous steps:
- assignment of all elements (structural parts, actors and documents) to a specific workflow
- assignment of all structural parts to a specific actor, according to lanes in BPMN
- assignment of all documents to particular tasks (as input or output)
- setting the order of all tasks or structural parts (e.g. Task1 is executed prior to Task2)

4 Demonstration of Feasibility

In this early stage of the project the ontology has been edited manually in Protégé. In future, most parts of the proposed procedure can be automated as all elements of a BPMN-workflow are transformable from XML to OWL-format according to transformation rules. Especially the most expensive steps 3 and 4 are highly automatable.

To be able to represent the relations between the workflow elements properly in Protégé, it is necessary to create them on instance level. Additionally, it is useful to organize them in a taxonomic structure (class hierarchy), for example similar or synonym tasks can be grouped in one class. This allows to capture and manage additional (implicit) knowledge. We decided to use OWL as a language because it containes a richer vocabulary for describing properties and classes compared to RDF. For example the representation of equivalent or disjoin classes, as well as transitivity of relations could be useful in our future research. For our tests of feasibility we used eight airport workflows for passenger and bagagge handling. Fig. 1. demonstrates an example of a BPMN-workflow.

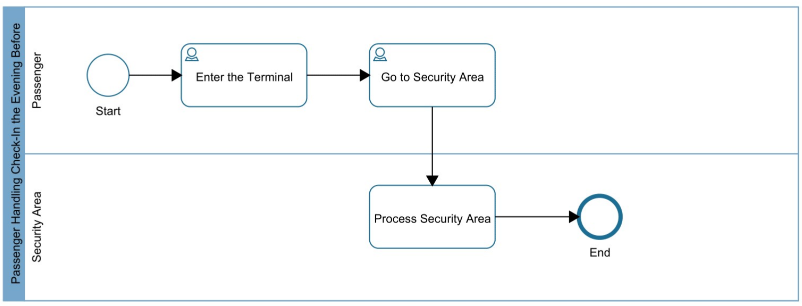


Fig. 1. Workflow 'Passenger Handling Check-In the Evening Before'

Following the four steps described in the previous section the workflow results in the ontology showed in Fig. 2. The continuous arrows demonstrate the hierarchy between the classes and instances. The dashed arrows stand for *ActivityPerformingRelations*, *TemporalRelations* and *AssignmentRelations*. According to this procedure all existing airport workflows are captured in the ontology. Based on the object properties it is possible to query and reassemble the initial workflows from the ontology, including the appropriate order of tasks. The ontology represents a case-base and serves as a knowledge base for further work.

Table 1. demonstrates some examples for transfering BPMN elements in OWL language. The left column depicts the typical BPMN diagrams or their parts and the right column contains appropriate OWL-code-snippets extracted from the ontology file.

The first line shows the assignment of workflow parts (tasks, events, actors or subprocesses) to a specific workflow with a particular name. In the workflow example in Fig. 1 all tasks, events and lanes (corresponding to actors) are part of a workflow with the name 'Passenger Handling Check-In the Evening Before'. The resulting OWL file is visualised in the ontology in Fig. 2. The node 'Passenger Handling Check-In the Evening Before' in the right bottom corner of Fig. 2 represents the workflow, which is connected to its elements via dashed lines.

The second line of the Table 1 stands for the assignment of workflow parts (tasks, gateways, events etc.) to a specific actor. This property captures the relationship 'who does what' or 'who is responsible for what'. In our workflow example in Fig. 1 the *Passenger* is the actor and thus responsible for the task *Enter the Terminal*. The third line shows the relationship between documents and tasks and illustrates if a document is an input or an output of a task.

Line four demonstrates the sequential order of workflow elements. For example, it depicts that Task 1 is executed prior to Task 2. In the line five of the table is an example of instance declaration. According to the step 3 of the procedure described in Section 3, all elements of a workflow are created as instances. As mentioned before, these instances can be ordered in a taxonomy.

5 Future Work

In the next project phase, we are going to automate the ontology construction procedure based on BPMN diagrams. The building of a hierarchical order in the ontology still requires an intervention of a domain expert. But there is a strong assumption that the construction of instances and the relations between them can be automated. We are aware of inaccuracies that can occur in the capturing of workflows, such as order of tasks (e.g. in one workflow task 1 proceeds prior to task 2 and in another workflow are those two tasks in a reversed order). Another problem can occur if two tasks are related in one workflow and in another workflow the same tasks have no relation at all. We need to find a way to handle these inaccuracies.

In some cases, the knowledge transfer from a source into the target domain is achievable only on a higher abstraction level. In the next step of our project we are going to pay attention to an automated abstraction of tasks. One promising approach is

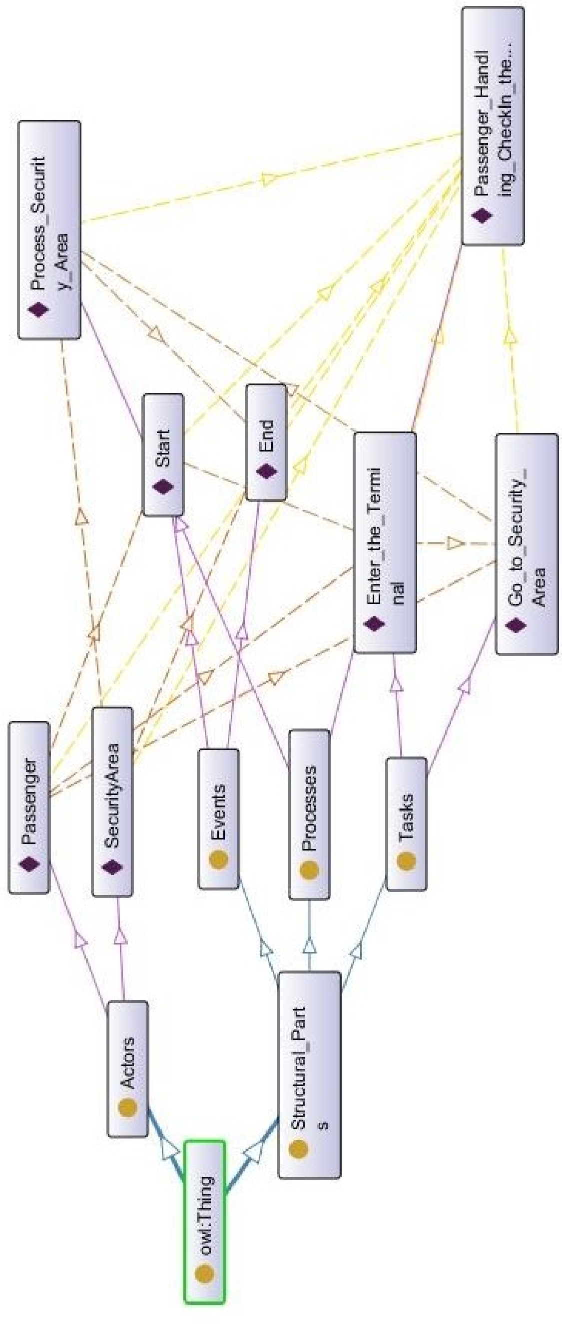


Fig. 2. Ontology of the workflow 'Passenger Handling Check-In the Evening Before'

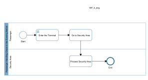
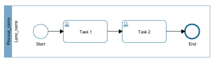
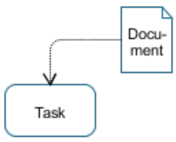
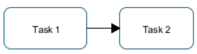
BPMN 2.0	**OWL-Code**
1. Assignment of structural parts, actors and documents to a particular workflow	
	<ObjectPropertyAssertion> <ObjectProperty IRI="#isPartOf"/> <NamedIndividual IRI="#Task/Event/Actor/Sub-Process"/> <NamedIndividual IRI="#Process_name"/> </ObjectPropertyAssertion>
2. Assignment of structural parts to a specific actor	
	<ObjectPropertyAssertion> <ObjectProperty IRI="#do"/> <NamedIndividual IRI="#Who/Lane_name"/> <NamedIndividual IRI="#Task/Event/Sub-Process"/> </ObjectPropertyAssertion>
3. Assignment of a document to a specific task	
	<ObjectPropertyAssertion> <ObjectProperty IRI="#isInput/OutputOf"/> <NamedIndividual IRI="#Document"/> <NamedIndividual IRI="#Task"/> </ObjectPropertyAssertion>
4. Setting of sequential order of tasks or events	
	<ObjectPropertyAssertion> <ObjectProperty IRI="#priorTo"/> <NamedIndividual IRI="#Task/Event/Sub-Process"/> <NamedIndividual IRI="#Task/Event/Sub-Process"/> </ObjectPropertyAssertion>
5. Creation of instances	
	<Declaration> <NamedIndividual IRI="#Task/Event/Gateway/Document"/> </Declaration>

Table 1. Examples of BMPN-elements in OWL

the decomposition in meaningful workflow streams and their abstraction. As stated in (Müller and Bergmann, 2014) the stream candidates must be identified, then stored in a repository and used for adaptation of workflows. We would like to examine the feasibility of abstracted workflow streams for knowledge transfer between the domains. (Müller and Bergmann, 2014) define conditions for building of meaningful workflow partitions. According to their specification, the tasks have to be transitively data-flow connected and each partition has to end with a creator task (task with a document as an output). Another condition is the data-flow completeness of a workflow (Müller and Bergmann, 2016). Our airport workflow examples contain only very few tasks with document output, so the conditions as stated in (Müller and Bergmann, 2014) and (Müller and Bergmann, 2016) can not be fulfilled. For our purpose, we need to define alternative boundaries for the separation of meaningful workflow streams and their abstraction.

Another topic in our project will be the investigation of a target domain. We plan to use the knowledge from the source domain (passenger and baggage handling) and transfer it to another area. The discussion of different potential target domains is still ongoing. In respect to a short distance transfer, we think about transferring workflow models between two different airports. A frequent situation in the airport service sector is the adjustment of processes after a change of the service provider. TL could contribute to optimize the existing workflows and, thus, support the airport quality management. Second, cargo processes may be a promising partner domain for passenger and baggage handling. In addition to transferring entire workflow models, we discuss to learn knowledge on exception handling routines for workflows (Reichert and Weber, 2012, ch. 6) in one domain and transfer them to the other domain. An example for such an exception handling in the two airport domains cargo and passenger/baggage handling is the compensating routine for flight delays, such as providing the customer with a voucher for beverages in case of a delay message from the electronic status reports of an airline.

Further, potential application fields might be the auditing of airport processes, their post-mortem analysis, or the recovery of normal operations after a disruption. We also would like to study transfer learning for more distant domains, e.g. the application of RFID technology in workflows in different industrial sectors.

6 Conclusions

In our work we proposed a new method for capturing process-oriented domain knowledge in an ontology. In Protégé, we recorded all parts of a workflow, first on an instance level and then in a taxonomical hierarchy. The taxonomy shows the relations between the elements of a workflow and contains further hierarchical knowledge. Additionally, we captured all relations between the elements of a BPMN workflow as properties, including the temporal relations. This ensures that any particular workflow can be restored from the ontology. We desribed the creation of the ontology in a procedure, which is generic and domain-independent. The ontology can be populated depending on new workflows added to the ontology. We outlined the usage of ontologies as a means for transfer learning, which of course is a topic of our further research.

References

[Abramowicz et al., 2012]Abramowicz, W., Filipowska, A., Kaczmarek, M., and Kaczmarek, T. (2012). Semantically enhanced business process modeling notation. In *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, pages 259–275. IGI Global.

[Baader et al., 2004]Baader, F., Horrocks, I., and Sattler, U. (2004). Description logics. In *Handbook on Ontologies*, pages 3–28. Springer.

[Falkenhainer et al., 1989]Falkenhainer, B., Forbus, K. D., and Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63.

[Fan et al., 2016]Fan, S., Hua, Z., Storey, V. C., and Zhao, J. L. (2016). A process ontology based approach to easing semantic ambiguity in business process modeling. *Data & Knowledge Engineering*, 102:57–77.

[Gruber, 2009]Gruber, T. (2009). Ontology. *Encyclopedia of database systems*, pages 1963–1965.

[Klenk et al., 2011]Klenk, M., Aha, D. W., and Molineaux, M. (2011). The case for case-based transfer learning. *AI Magazine*, 32(1):54–69.

[Klenk and Forbus, 2013]Klenk, M. and Forbus, K. (2013). Exploiting persistent mappings in cross-domain analogical learning of physical domains. *Artificial intelligence*, 195:398–417.

[Könik et al., 2009]Könik, T., O'Rorke, P., Shapiro, D., Choi, D., Nejati, N., and Langley, P. (2009). Skill transfer through goal-driven representation mapping. *Cognitive Systems Research*, 10(3):270–285.

[Kudenko, 2014]Kudenko, D. (2014). Special issue on transfer learning. *KI-Künstliche Intelligenz*, 1(28):5–6.

[Kuhlmann and Stone, 2007]Kuhlmann, G. and Stone, P. (2007). Graph-based domain mapping for transfer learning in general games. In *European Conference on Machine Learning*, pages 188–200. Springer.

[Matt et al., 2015]Matt, C., Hess, T., and Benlian, A. (2015). Digital transformation strategies. *Business & Information Systems Engineering*, 57(5):339–343.

[Minor et al., 2014a]Minor, M., Bergmann, R., and Görg, S. (2014a). Case-based adaptation of workflows. *Information Systems*, 40:142–152.

[Minor et al., 2016]Minor, M., Bergmann, R., Müller, J.-M., and Spät, A. (2016). On the Transferability of Process-Oriented Cases. In Goel, A. K., Díaz-Agudo, M. B., and Roth-Berghofer, T., editors, *Case-Based Reasoning Research and Development - 24th International Conference, ICCBR 2016, Atlanta, GA, USA, October 31 - November 2, 2016, Proceedings*, volume 9969 of *Lecture Notes in Computer Science*, pages 281–294. Springer.

[Minor et al., 2014b]Minor, M., Montani, S., and Recio-García, J. A. (2014b). Editorial: Process-oriented case-based reasoning. *Inf. Syst.*, 40:103–105.

[Montani and Leonardi, 2014]Montani, S. and Leonardi, G. (2014). Retrieval and clustering for supporting business process adjustment and analysis. *Information Systems*, 40:128–141.

[Müller and Bergmann, 2014]Müller, G. and Bergmann, R. (2014). Workflow streams: a means for compositional adaptation in process-oriented cbr. In *International Conference on Case-Based Reasoning*, pages 315–329. Springer.

[Müller and Bergmann, 2015]Müller, G. and Bergmann, R. (2015). Generalization of workflows in process-oriented case-based reasoning. In *FLAIRS Conference*, pages 391–396.

[Müller and Bergmann, 2016]Müller, G. and Bergmann, R. (2016). Case completion of workflows for process-oriented case-based reasoning. In *International Conference on Case-Based Reasoning*, pages 295–310. Springer.

[Natschläger, 2011]Natschläger, C. (2011). Towards a bpmn 2.0 ontology. In *International Workshop on Business Process Modeling Notation*, pages 1–15. Springer.

[Pan and Yang, 2010]Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.

[Ragni and Strube, 2014]Ragni, M. and Strube, G. (2014). Cognitive complexity and analogies in transfer learning. *KI-Künstliche Intelligenz*, 28(1):39–43.

[Reichert and Weber, 2012]Reichert, M. and Weber, B. (2012). *Enabling flexibility in process-aware information systems: challenges, methods, technologies*. Springer Science & Business Media.

[Richter and Weber, 2016]Richter, M. M. and Weber, R. O. (2016). *Case-based reasoning*. Springer.

[Rospocher et al., 2014]Rospocher, M., Ghidini, C., and Serafini, L. (2014). An ontology for the business process modelling notation. In *FOIS*, pages 133–146.

[Taylor and Stone, 2009]Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685.

[Thomas and Fellmann, 2009]Thomas, O. and Fellmann, M. (2009). Semantic process modeling – design and implementation of an ontology-based representation of business processes. *Business & Information Systems Engineering*, 1(6):438–451.