Ontology-based transfer learning in the airport and warehouse logistics domains

Miriam Herold and Mirjam Minor

Goethe University, Frankfurt am Main 60325, Germany
http://www.wi.cs.uni-frankfurt.de
{herold,minor}@cs.uni-frankfurt.de

Abstract. This work is a position paper for the examination of ontology-based transfer learning in the context of business processes. We continue our preliminary work on transfering process-oriented knowledge from a well-known source domain to a less specified target domain. We outline our ideas on workflows from two specific contexts: passenger and baggage logistics at the airport on one hand and warehouse management logistics on the other hand. In the first step we automatically transform BPMN files from these two domains in two separate ontologies. In the next step we intend to use ontology mapping as a means for the transfer. We plan to examine the concepts of generalization and abstraction to ease the transfer. We claim that the mentioned domains are feasible candidates for transfer learning, as we find several analogies between the airport handling and warehouse management workflows. Additionally we discuss possible utilization resp. benefits of the transfer learning within this two particular domains and draft the next steps for the future research.

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

1 Introduction

Transfer learning - as a method for using the knowledge from one well-known domain and adapting and re-using it in another one [4] - is currently a very relevant topic in different research contexts, mainly in data mining [13] or machine learning [19], [7]. Neural networks can benefit from transfer learning, as pretrained networks can be adapted to another related tasks and reduce significantly the training costs [12].

Aside from the scientific world there are manyfold practice-oriented areas, where transfer learning can offer a considerable support. One of them is the management of business processes. Many enterprises face the challenge of digital transformation. Digitization not only affects production, but also processes, management concepts and the whole organization structure [8]. The changes require a reasonable level of automation as well as flexibility in modeling and adaptation of workflows. Modeling business processes from scratch usually binds resources and can cause substantial costs. Sometimes enterprises simply suffer from lack of expertise and workflows can not be modelled in an appropriate quality. At this point *case-based reasoning* (CBR) methods might support the effort of an enterprise. CBR is based on the intuition that similar problems tend to have similar solutions [16]. Already modelled business processes represent experiential knowledge and, after an adaption, they can be re-used in modeling of new workflow models. In the context of *process-oriented case-based reasoning* (POCBR), particular process model can be seen as a *solution* for the *problem* of reaching a certain outcome, completing a set of activities, typically described as a sequence of tasks with a clear assignment of responsibilities. This problem-solution pairs are *cases* and build a *case-base*, which can be queried on the basis of specified features of a new reqested process model. Workflows retrieved from the case-base can represent a solution for the new upcoming problem, usually after an adaptation to the specific needs of the new request.

In the past decade reasonable amount of research was done in POCBR. But there is still less literature about using transfer learning methods for transfering process-oriented knowledge. In our project EVER2[1] we force these ideas and explore the transferability of workflows. According to our previous research there is an evidence that workflows are transferable from one well-known domain into another domain, where the knowledge is sparse. We are convinced that the transfer is possible and also meaningful, if there is some overlap between the domains and the target domain can benefit from the transferred workflows in various ways [9]. In our work we examine ontologies as a means for the transfer, since they are a suitable tool for describing relations between the parts of a workflow and allow ontology mapping. Additionaly, we intend to accommodate the concepts of abstraction and generalization for workflows in an ontology. These concepts are able to ease the transfer as stated in [9].

Workflows often consist of very domain-specific tasks. When transfering into the target domain in the initial form, it might be difficult to find correspondencies. Generalization can ease the problem without changing the original structure of a workflow. We assume that if the descriptions of activities are replaced with another, more general terms, it will facilitate the finding of analogies with the target domain. After the transfer, generalized workflows have to be specified to the original form and usually also adapted according to the needs of the target domain. We plan to examine, if a developed hierarchy of tasks can be used for automated generalization of workflows and if the method utilizes transfer learning, as stated in [9].

In this paper we introduce two domains as possible candidates for transfering process models. One domain is *passenger and baggage handling at the airport* and the second one is *warehouse management*. Our hypothesis for this position

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

paper is that the two introduced domains are feasible candidates for transfer learning. We transformed the available workflows from this two domain repositories into the BPMN 2.0 format. In the past years BPMN became de facto standard in modeling of business processes and the underlying XML format provides a good base for further automation in processing of workflows. It allows a good transformation in an ontology as well as retrieval without any losses of information. The steps of building a domain-independent ontology from BPMN workflows are matter of our previous publication [3].

This paper is structured as follows: In the next section we provide an overview about the related work. Third section describes the data foundation and where we currently stand in our project. Section four offers a proposal for further development of transfer learning methods, such as generalization and abstraction. In section five we discuss the mentioned domains as potential transfer learning use cases and motivate, why we believe they are good candidates for transfering workflows. At the end we summarize our paper and draw some conlusions.

2 Related Work

Up to now there has been done a considerable amount of research on case-based reasoning. Generally speaking, CBR systems contain problem-solution pairs (cases), which are collected in a case-base. The cases can be retrieved, adapted and re-used for new upcoming problems. If the suggested solution is able solve the new problem, CBR system adds the new problem-solution pair into the case-base. If not, the retrieval result will be rejected and the system stores this information as well. Due to this learning ability some authors describe CBR as a transfer learning method [4]. Our research is a contribution to this topic and we extend it to procedural knowledge. First literature on transferability of process-oriented cases indicates that workflows are transferable [9]. We plan to enhance this efforts and automate transfering of procedural knowledge.

So far there is still little research on transfer learning for process-oriented case-based reasoning (POCBR). Our goal is to examine POCBR methods for transfering process knowledge from one domain into another one. One of the valuable methods are analogical models. Analogy can be used for different kind of scientific questions on transfer learning and there is a substantial amount of literature on analogical models [1], [6], [14], [5]. To be able to transfer procedural knowledge there has to exist some overlap between the source and the target domain. This overlap can be represented by analogy. Therefore, one of the directions for our research is finding analogies between workflows, or at least between their parts.

In the literature there are some approaches for building of process-oriented ontologies, as for example the POBA-approach [2]. Compared to our goal of transfering workflows, their idea is to disambiguate and improve the quality in modeling of workflows. There also exists literature on BPMN-ontologies [11], [17]. They focus on capturing all BMPN elements and the relations between them. As we only use a small part of the BPMN vocabulary, we decided not to utilize a whole BPMN ontology but only successively add the required elements.

3 Data Foundation

In this section we would like to describe the current state of our project and the preliminary results along with some data foundations. As mentioned in the introductory section, we propose two diffrent domains for transfer learning in business processes. The first domain is *passenger and baggage handling at the airport*. Second domain is *SAP warehouse management*. Below we show some typical workflow examples and snippets from the ontologies built for the two domains. We used *Camunda Modeler* as a modeling tool and generated in total 50 XML files. For creating ontologies we used *Protégé*. It is a publicly available tool and so far it covers all functionalities we need for our purposes.

For the repository in the airport passenger and baggage handling domain we extracted 30 workflows, mainly different kinds of passenger check-in, baggage handling, security checks (for passengers and baggage), customs clearance and various workflows for transport and loading of baggage into the aircraft. Most of the workflows are based on a textbook [15] and modelled in BMPN 2.0. Fig. 1 shows an example of a workflow from the airport domain.

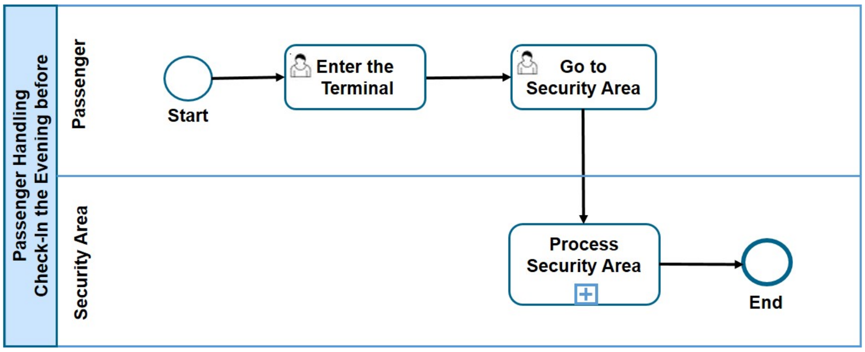


Fig. 1. Workflow 'Passenger Handling Check-In the Evening Before'

In the SAP warehouse management domain we extracted 20 different workflows from the SAP website [18]. The website is a collection of best practices for integration of SAP Extended Warehouse Management and SAP S/4HANA rapid deployment solution. It contains different process models, primarily designed for the integration of SAP modules, but they offer a solid amount of business procedures to extract workflows with the appropriate control flow. The processes cover e.g. inbound and outbound of products in/from various kinds of warehouses, replenishment, scrapping, inventory as well as consumption of products during production. We also transformed these workflows into BPMN. As we found some inconsistencies in the proprietary format, it was necessary to define rules for modeling them in BPMN. Fig. 2 shows a snippet of a workflow from the warehouse management domain.

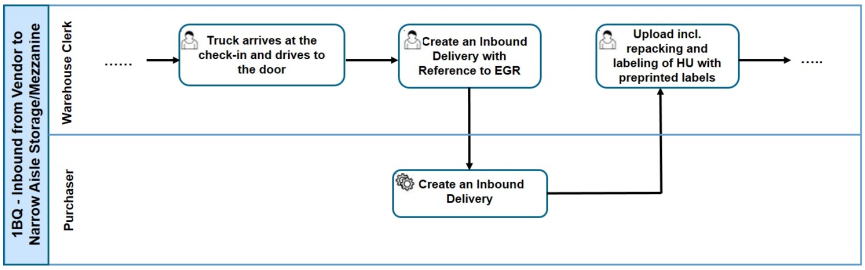

1BQ - Inbound from Vendor to Narrow Aisle Storage/Mezzanine
Warehouse Clerk
Purchaser
......
Truck arrives at the check-in and drives to the door
Create an Inbound Delivery with Reference to EGR
Upload incl. repacking and labeling of HU with preprinted labels
.....
Create an Inbound Delivery

Fig. 2. Part of the workflow 'Inbound from Vendor to Narrow Aisle Storage/Mezzanine'

In the next step of our project we created ontologies for both domains. Their development follows the instructions according to our previous publication [3]. The steps capture all elements and relations of a workflow, so it is completely covered in an ontology. According to this procedure it is possible to create workflow ontologies independent of any underlying domain. In Protégé we created basic classes for BPMN elements and also some basic properties to be able to cover all necessary relations. In our work we don't focus on covering the entire BPMN vocabulary. We are aware that there are existing BPMN ontologies, but for the sake of simplicity we only add elements, that really occurs in our workflows.

As next, we imported workflows into the ontology in an automated manner and developed a JAVA-based tool for this purposes. Using a DOM parser we are able to transform all XML files into an OWL format and create a workflow ontology according to the predefined rules. We evaluated this procedure and developed a SPARQL filter for retrieving particular workflows from an ontology. The results show, that we are able to retrieve a workflow from the ontology wihtout any loss of information. The query filters out all elements assigned to a workflow, e.g. lanes, actors, tasks and also the control flow, so the initial workflow can be completely restored, if necessary. In Fig. 3 we demonstrate a result of the SPARQL query for the the workflow drawn in Fig. 1, after was added and retrieved from an ontology. The sample does not contain any documents or gateways, but of course they can be restored as any other part of a workflow.

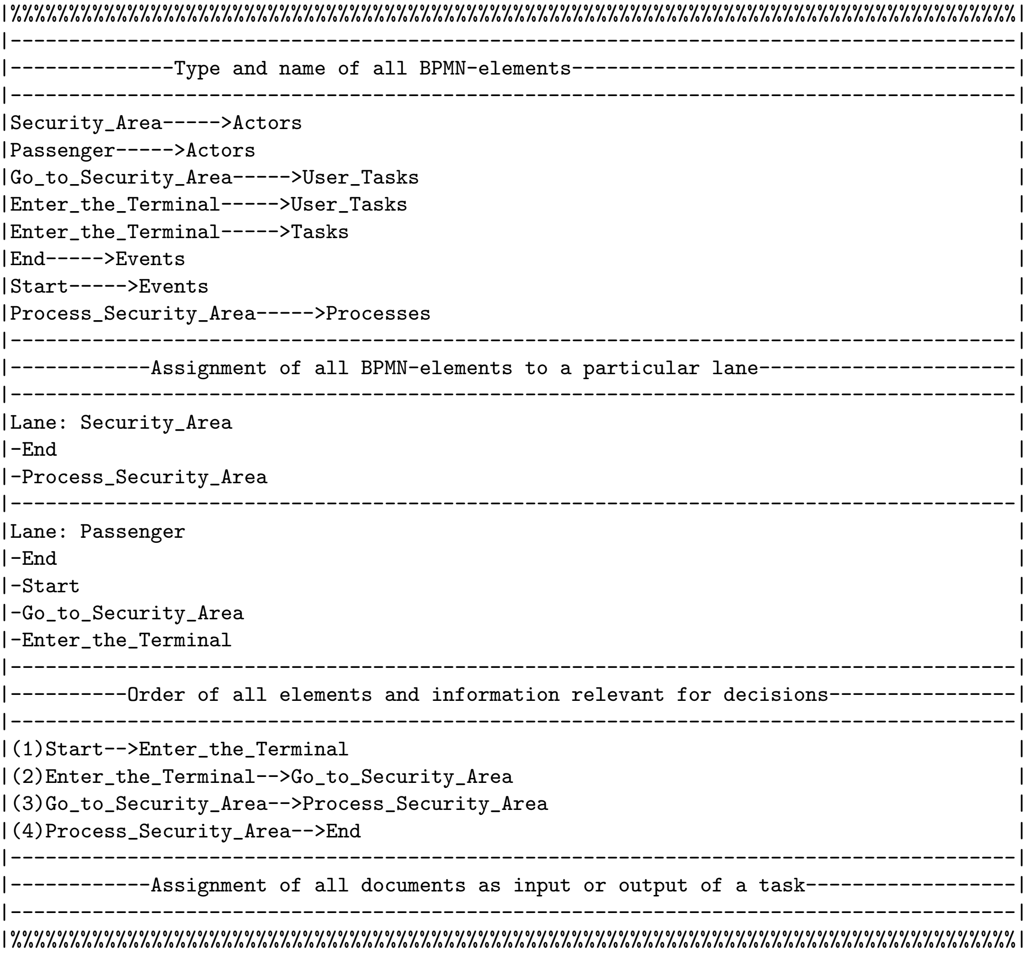
Another contribution to our project is building of a task hierarchy in the airport domain. The reason why we created the hierarchy is the idea of generalization of workflows. In the next section we will go into this in more detail, and explain how generalization can be used in transfer learning. During the modeling of the airport workflows, we realized that there are various tasks with a very similar meaning, e.g. movement of pieces of luggage. Baggage can be moved by the airport staff, baggage conveyor system or transported in dollies or containers through the airport. These tasks can be sorted in classes, depending on the type of the movement. Building the task hierarchy still needs to be done manually. At this stage of our project, the tool wouldn't be able to recognize the exact meaning of tasks yet and group them automatically in classes. In total we have about 125 different tasks and they are classified in four different hierarchy levels. The hierarchy can easily be added to the airport ontology and serves as a basis for the generalization of workflows. Fig. 4 demonstrates a small excerpt of the task hierarchy for the airport domain.

4 Further development of methods for the transfer learning process

Transfer learning is useful in cases where workflows can not be transferred directly in the original form. If there is a relevant overlap between the domains, workflows can be transferred in the original structure, but by using of more general terms. In our previous work we already showed on manual basis that generalization of workflows can ease the identification of correspondencies [9]. In case the overlap between the domains is considerably smaller, the ideas of analogical mapping can be considered as a means for the transfer.

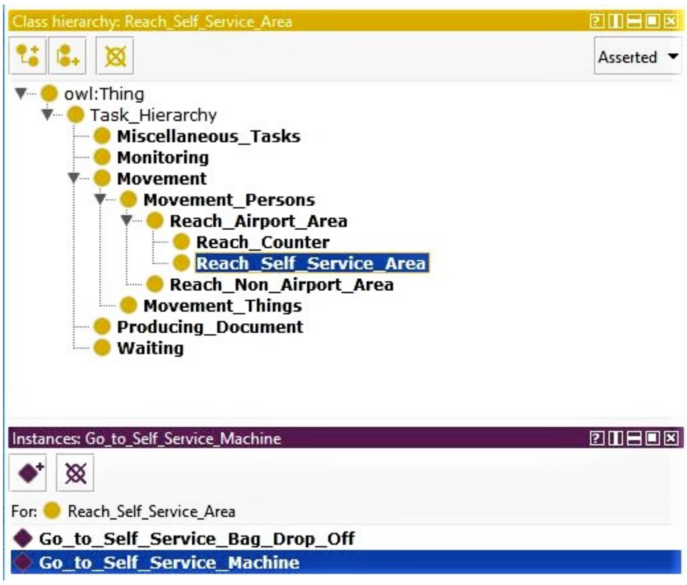
We plan to extend our previous work and automate the creation of generalized workflows. During the phase of modeling in BPMN we found several analogies between the tasks in both domains, e.g. movement of goods, check-in procedure etc. For the aiport domain we already developed a task hierarchy and accommodated it into the ontology. We intend to build a task hierarchy and then produce generalized workflows also for the warehouse domain. Up to now we are able to capture all necessary relations of a workflow in an ontology. So we are optimistic about using ontologies for the automated generalization of workflows. Once a generalized workflow is transferred, it has to be specified and/or adapted according to the requirements of the target domain.

Generalization does not change the original structure of a workflow. The business processes in our repositories indicate that some of the workflows might be too complex for the transfer via generalizatio only. According to the previous



```
|%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
|--------------------------------------------------------------------------------------|
|--------------Type and name of all BPMN-elements--------------------------------------|
|--------------------------------------------------------------------------------------|
|Security_Area----->Actors                                                             |
|Passenger----->Actors                                                                 |
|Go_to_Security_Area----->User_Tasks                                                   |
|Enter_the_Terminal----->User_Tasks                                                    |
|Enter_the_Terminal----->Tasks                                                         |
|End----->Events                                                                       |
|Start----->Events                                                                     |
|Process_Security_Area----->Processes                                                  |
|--------------------------------------------------------------------------------------|
|------------Assignment of all BPMN-elements to a particular lane----------------------|
|--------------------------------------------------------------------------------------|
|Lane: Security_Area                                                                   |
|-End                                                                                  |
|-Process_Security_Area                                                                |
|--------------------------------------------------------------------------------------|
|Lane: Passenger                                                                       |
|-End                                                                                  |
|-Start                                                                                |
|-Go_to_Security_Area                                                                  |
|-Enter_the_Terminal                                                                   |
|--------------------------------------------------------------------------------------|
|----------Order of all elements and information relevant for decisions----------------|
|--------------------------------------------------------------------------------------|
|(1)Start-->Enter_the_Terminal                                                         |
|(2)Enter_the_Terminal-->Go_to_Security_Area                                           |
|(3)Go_to_Security_Area-->Process_Security_Area                                        |
|(4)Process_Security_Area-->End                                                        |
|--------------------------------------------------------------------------------------|
|------------Assignment of all documents as input or output of a task------------------|
|--------------------------------------------------------------------------------------|
|%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
```

Fig. 3. Screenshot of the retrieved workflow drawn in Fig. 1



Class hierarchy: Reach_Self_Service_Area

Asserted

- owl:Thing
  - Task_Hierarchy
    - Miscellaneous_Tasks
    - Monitoring
    - Movement
      - Movement_Persons
        - Reach_Airport_Area
          - Reach_Counter
          - Reach_Self_Service_Area
        - Reach_Non_Airport_Area
      - Movement_Things
    - Producing_Document
    - Waiting

Instances: Go_to_Self_Service_Machine

For: Reach_Self_Service_Area

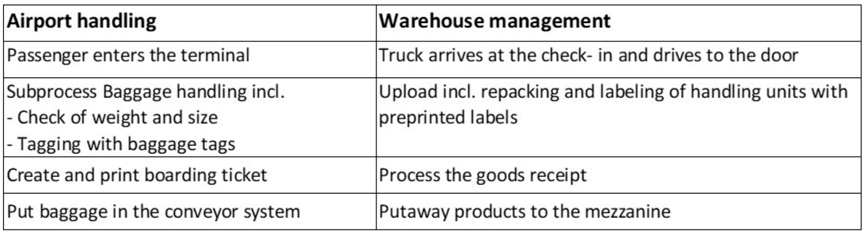
- Go_to_Self_Service_Bag_Drop_Off
- Go_to_Self_Service_Machine

Fig. 4. Part of the task hierarchy in Protégé

work on transferability of process-oriented cases, worklows are better transferable on a higher abstraction level [9]. At this point the method of *compositional adaption* could help to reduce the complexity of workflows and help to find analogies. One way to realize compositional adaptation for workflows is to decompose them into meaningful parts, called *workflow streams* [10]. A workflow stream can be represented and replaced by an abstract task, so it offers an immediate method for abstraction [9]. The rules for identifying of workflow streams according to the existing literature [10] are mostly based on documents consumed or produced by tasks. The workflows in our two domains are less document-driven, so we need to define another rules for identifying of workflow parts suitable for abstraction. We intend to accommodate this rules and also the abstracted workflows in the ontology. To make transferred workflows executable, the abstraction has to be reversed and the workflows adapted in the target domain.

5 Potential transfer learning use cases and benefit discussion

In this section we would like to outline, why we believe the proposed two domains are feasible candidates for transfer learning. During the workflow modeling we detected analogies in tasks and partially also in the control flow. Fig. 5 shows some examples of these analogies. Left column describes tasks from check-in and baggage handling processes at the airport. In most cases these tasks are executed in the same order as in the table. Right column represents their correspondencies in the warehouse domain. It illustrates the workflow snippet drawn in Fig. 2. It is obvious that both, the tasks themselves as well as the order of tasks are quite similar. This is a clear indicator that the two domains are extremely promising for further examination of transfer learning.



Airport handling	**Warehouse management**
Passenger enters the terminal	Truck arrives at the check- in and drives to the door
Subprocess Baggage handling incl. - Check of weight and size - Tagging with baggage tags	Upload incl. repacking and labeling of handling units with preprinted labels
Create and print boarding ticket	Process the goods receipt
Put baggage in the conveyor system	Putaway products to the mezzanine

Fig. 5. Example of analogical tasks from both domains

At this point we sketch some ideas for possible applications of workflow transfer between the introduced domains. We are convinced that workflows from one domain can support the *workflow modeling* in the other domain. Sometimes enterprises suffer from lack of expertise or the modelers need instant help in creating workflow models. Transferred workflows or at least parts of them can be used as a template and help in the creation of models in the target domain, given the sufficient overlap.

Transferred workflows can also be used as additional knowledge for *exception handling*. For instance when after a flight delay the passenger misses the connecting flight. Instead of transiting his/her baggage from one aircraft into the connecting one directly on the airfield, it has to be checked out and later checked in. For all these scenarios we posses workflow models in our repository. A similar situation can occur in the warehouse domain. Many warehouses serve as distribution centers and are spread in a star-shape through the countries. They usually do not store goods but load them from one truck into another one for further distribution. Once a truck has a delay and the connecting one can not wait, the goods have to be stored until further distribution. This requires exception handling, such as inbound and outbound of products. We believe that with the workflows from the airport domain we would be able to handle this exception and model or apply addapted workflows in the warehouse domain.

Another potential use case might be the enrichment with sensory capabilities, for example based on RFID technology. Airports normally use bar code sensoring. Baggage tags contain bar codes that can be read out at different points of the airport. This technology is quite limited as bar codes are not able to store as much of the information as RFID codes. Airport operators are aware of this limitations but they often hesitate to implement RFID technology as it is very cost-intensive. Compared to this, in the logistics and warehouse domain RFID technology is widespread. RFID codes can store extensive information, e.g. observation of the cold chain etc. Potentially, experiences from utilizing RFID in the logistics and warehouse domain might be transferred to the aiport domain, and help for example with finding of best positioning for RFID scanners. But at this stage of our project, this idea is only a skech and a matter of further examination.

6 Conclusion

In this paper we describe the progress of our project on transfer learning for case-based reasoning. We gained two workflow repositories from the domains airport passenger/baggage handling and warehouse management in BPMN format. We developed tools for transforming these workflow models into ontologies and also for retrieving of workflow models from an ontology. We created a task hierarchy in one of those two domains as a base for workflow generalization. Then we outline the next steps for the research on using ontologies for generalization and abstraction. We explain the background of these methods and outline how we plan to utilize them for our purposes. We have elaborated sample analogies that are a strong indicator that our hypothesis can be confirmed and the two proposed domains provide valuable use cases for transfer learning. We introduce some potential application scenarios for transfer learning and discuss why we believe the introduced domains are feasible candidates for transferring workflows.

Acknowledgement

We would like to thank Jan Höcher for his support in the development of tools described in the third section.

References

1. B. Falkenhainer, K. D. Forbus, and D. Gentner. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63, 1989.
2. S. Fan, Z. Hua, V. C. Storey, and J. L. Zhao. A process ontology based approach to easing semantic ambiguity in business process modeling. *Data & Knowledge Engineering*, 102:57–77, 2016.
3. M. Herold and M. Minor. Ontology-based representation of workflows for transfer learning. In *Proceeding of the Conference "Lernen, Wissen, Daten, Analysen", Mannheim, Germany, August 22-24, 2018*, pages 139–149, 2018.
4. M. Klenk, D. W. Aha, and M. Molineaux. The case for case-based transfer learning. *AI Magazine*, 32(1):54–54, 2011.

5. M. Klenk and K. Forbus. Exploiting persistent mappings in cross-domain analogical learning of physical domains. *Artificial intelligence*, 195:398–417, 2013.
6. T. Könik, P. O'Rorke, D. Shapiro, D. Choi, N. Nejati, and P. Langley. Skill transfer through goal-driven representation mapping. *Cognitive Systems Research*, 10(3):270–285, 2009.
7. D. Kudenko. Special issue on transfer learning. *KI - Künstliche Intelligenz*, 28:5–6, 02 2014.
8. C. Matt, T. Hess, and A. Benlian. Digital transformation strategies. *Business & Information Systems Engineering*, 57(5):339–343, 2015.
9. M. Minor, R. Bergmann, J.-M. Müller, and A. Spät. On the Transferability of Process-Oriented Cases. In A. K. Goel, M. B. Díaz-Agudo, and T. Roth-Berghofer, 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, 2016.
10. G. Müller and R. Bergmann. Generalization of workflows in process-oriented case-based reasoning. In *The Twenty-Eighth International Flairs Conference*, 2015.
11. C. Natschläger. Towards a bpmn 2.0 ontology. In *International Workshop on Business Process Modeling Notation*, pages 1–15. Springer, 2011.
12. M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.
13. S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
14. M. Ragni and G. Strube. Cognitive complexity and analogies in transfer learning. *KI-Künstliche Intelligenz*, 28(1):39–43, 2014.
15. A. Richter. *Gepäcklogistik auf Flughäfen: Grundlagen, Systeme, Konzepte und Perspektiven*. Springer-Verlag, 2012.
16. M. M. Richter and R. O. Weber. *Case-based reasoning*. Springer, 2016.
17. M. Rospocher, C. Ghidini, and L. Serafini. An ontology for the business process modelling notation. In *FOIS*, pages 133–146, 2014.
18. SAP. Sap best practices explorer, 2019 (accessed July 7, 2019). `https://rapid.sap.com/bp/#/browse/categories/sap_s%254hana/areas/on-premise/packageversions/RDS_EWM_S4`.
19. M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.