Ontology-based Transfer Learning in the Airport and Warehouse Logistics Domains

Miriam Herold[a], Mirjam Minor[a]

[a]*Goethe-University Frankfurt, Robert-Mayer-Str. 10, Frankfurt am Main, 60325, Germany*

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 transferring 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 [1] - is currently a very relevant topic in different research contexts, mainly in data mining [2] or machine learning [3], [4]. Neural networks can benefit from transfer learning, as pretrained networks can be adapted to another related tasks and reduce significantly the training costs [5].

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 [6]. 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 [7]. Already modelled business processes represent experiential knowledge and, after an adaption, they can be re-used

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, March 22-24, 2021.
herold@informatik.uni-frankfurt.de (M. Herold); minor@informatik.uni-frankfurt.de (M. Minor)
http://www.wi.informatik.uni-frankfurt.de/ (M. Herold); http://www.wi.informatik.uni-frankfurt.de/ (M. Minor)
0000-0002-9139-8610 (M. Herold); 0000-0002-6592-631X (M. Minor)

in modeling of new workflow models. In the context of *process-oriented case-based reasoning* (POCBR), a 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. These 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 adaption 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 transferring process-oriented knowledge across the boarders of application domains. 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 [8]. 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 [8].

Workflows often consist of very domain-specific tasks. When transferring into the target domain in the initial form, it might be difficult to find correspondences. 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 supports transfer learning, as stated in [8]. Acquiring analogical knowledge in a computational approach is a challenging issue. In one of our previous publications we presented a solution that combines learning with knowledge engineering. [9] Our approach is to use word embeddings for learning analogies in workflow tasks.

In this paper we introduce two domains as possible candidates for transferring process models. This work is an extension of a workshop paper presented at the ICCBR conference in 2019 in Otzenhausen [10]. One proposed domain is *passenger and baggage handling at the airport* and the second one is *warehouse management*. Our hypothesis for this position paper is that the two introduced domains are feasible candidates for transfer learning. In the first step we transformed the available workflows from this two domain repositories into the BPMN 2.0 format. In the past decade 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 transformation in an ontology as well as retrieval without losing valuable information. The steps of building a domain-independent ontology from BPMN workflows are matter of our previous publication [11].

The remainder of the 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

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

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 transferring 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, the 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 [1]. 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 [8]. We plan to enhance this efforts and automate transferring 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 transferring 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 [12], [13], [14], [15]. 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 [16]. Compared to our goal of transferring workflows, their idea is to disambiguate and improve the quality in modeling of workflows. There also exists literature on BPMN-ontologies [17], [18]. 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*[2] as a modeling tool and generated in total 50 XML files. For creating ontologies we used *Protégé*[3]. It is a publicly available tool and so far, it covers all functionalities needed for our purposes.

[2]https://camunda.com/de/products/camunda-bpm/modeler/
[3]https://protege.stanford.edu/

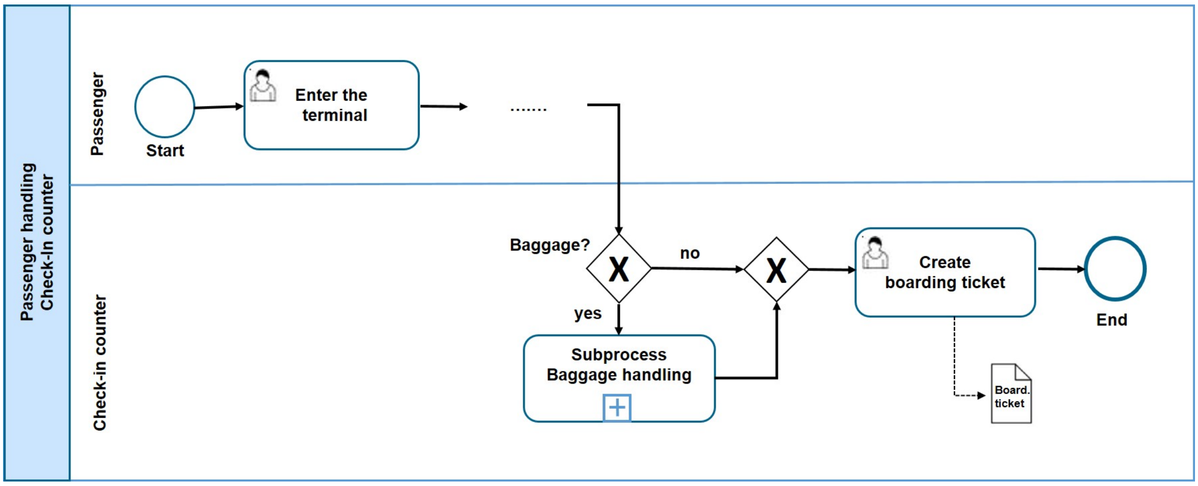

Figure 1: Workflow 'Passenger handling at check-in counter'

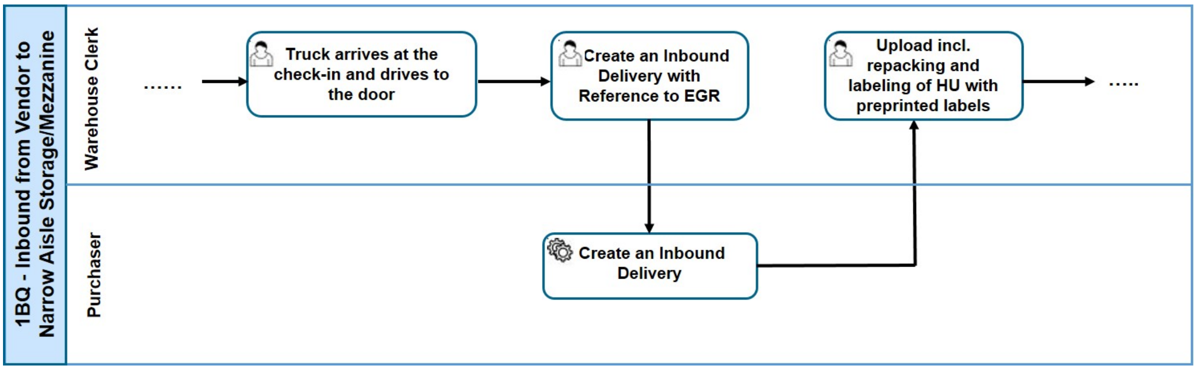

Figure 2: Part of the workflow 'Inbound from vendor to narrow aisle storage/mezzanine'

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 [19] and modelled in BMPN 2.0. Fig. 1 shows an example of a partial workflow from the airport domain.

In the SAP warehouse management domain we extracted 20 different workflows from the SAP website [20]. 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.

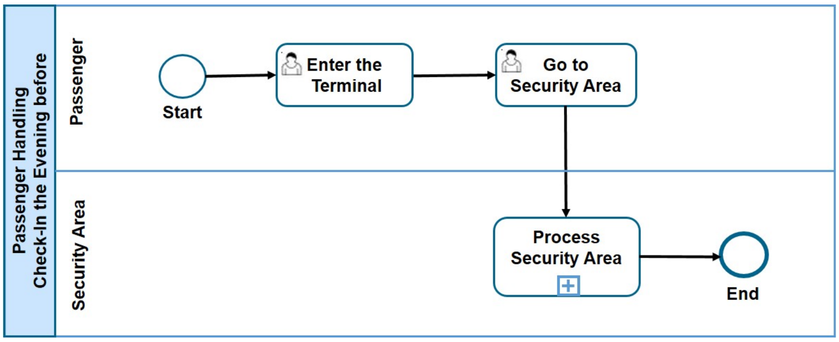

Figure 3: Workflow 'Passenger handling check-in the evening before'

In the next step of our project, we created ontologies for both domains. Their development follows the instructions according to our previous publication [11]. 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 address 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 purpose. With 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 without losing 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. 4 we demonstrate a result of the SPARQL query for the the workflow drawn in Fig. 3, after it was added into and retrieved from an ontology. The sample does not contain any documents or gateways, but they easily 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 ordered in classes, depending on the type of the movement. Currently, building of the task hierarchy still needs to be done manually. At this stage of our project, the tool would not 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 serve as a basis for the generalization of workflows. Fig. 5 demonstrates a small excerpt of the task hierarchy for the airport domain.

```
|%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
|--------------------------------------------------------------------------------|
|--------------Type and name of all BPMN-elements--------------------------------|
|--------------------------------------------------------------------------------|
|Security_Area----->Actors                                                       |
|Passenger----->Actors                                                           |
|Go_to_Security_Area----->User_Tasks                                             |
|Enter_the_Terminal----->User_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                                                                 |
|-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-------------|
|--------------------------------------------------------------------------------|
|%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
```

Figure 4: Screenshot of the retrieved workflow drawn in Fig. 3

4. Further development of methods for the transfer learning process

Transfer learning (TL) addresses the question of “how the things that have been learned in one context can be re-used and adapted in related contexts” [4]. The source domain D_S represents the context where knowledge is available at a mature level. The target domain D_T provides a context where the knowledge is sparse. In TL for POCBR, the procedural knowledge to be transferred is contained in workflows. D_S comprises a large repository of workflows called WF_S. The workflow repository WF_T in the target domain is small and to be enriched by a set of transferred workflows denoted by $WF_{T'}$. As depicted in Fig. 6, the transfer process is performed by using transfer strategy based on ontological knowledge and transfer rules. The user is a workflow designer who aims to model workflows in D_T. The transfer process creates entire workflows or parts of workflows in WF_T from workflows in WF_S, which are approved and further developed by the user. The methods of generalization, abstraction and analogical substitution provide the translation rules to transform workflows across the domains. The order of tasks remains unchanged during the transfer 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 correspondences [8]. In case the overlap between the domains is considerably smaller, the ideas of analogical mapping can be taken into account as a means for the transfer.

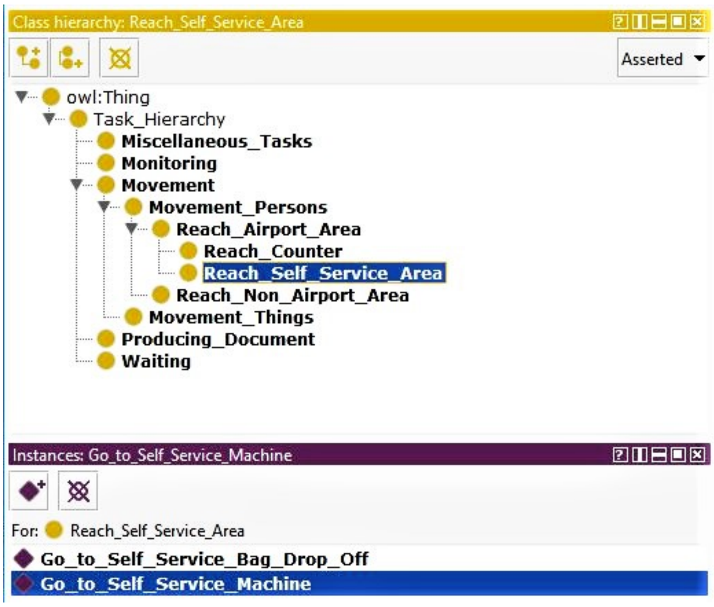

Figure 5: Part of the task hierarchy in Protégé

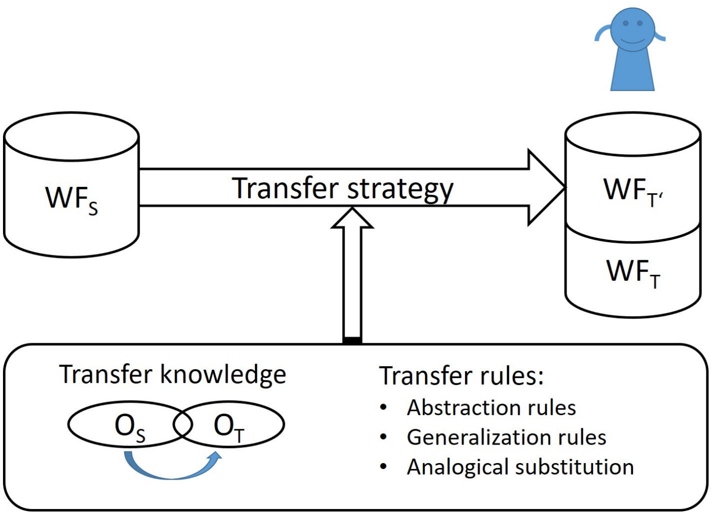

Figure 6: Strategy for transferring of workflows

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 both domains, we already developed task hierarchies and accommodated them into the ontologies. 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.

Business processes in our repositories indicate that some of the workflows might be too complex

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

Figure 7: Example of analogical tasks from both domains

for the transfer via generalization only, as it does not change the original structure of a workflow. According to the previous work on transferability of process-oriented cases, worklows are better transferable on a higher abstraction level [8]. 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* [21]. A workflow stream can be represented and replaced by an abstract task, so it offers an immediate method for abstraction [8]. The rules for identifying workflow streams according to the existing literature [21] 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 other rules for identifying workflow parts suitable for abstraction. More promising to resolve this issue seems to be the approach introduced by Polyvyanyy et al. [22]. It contains different abstraction rules and the result are workflows with a less complex structure. Our first experiments show that it is possible to perform the abstraction automatically. The abstracted workflows can be added into the ontology with a reference to the original workflow model. 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. 7 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. Right column represents their correspondences 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. We are aware that in many domains there exist reference models. Whenever available, they should be used preferably for modeling new workflows. Our aim is to find a transfer method that can be used in domains with no available reference model.

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 offer additional knowledge for *exception handling*. For instance, in cases when passenger misses his/her connecting flight due to a flight delay. Instead of transiting the 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/handling units. 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 possible scenario might be the Covid-19 vaccination centers. They need logistics procedures as well as handling of the person flow.

The next 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 airport points. 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 the 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 have described 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 task hierarchies in both domains as a base for workflow generalization. Then we outlined the next steps for the research on using ontologies for generalization and abstraction. We explained the background of these methods and outlined how we plan to utilize them for our purposes. We have elaborated sample analogies as a strong indicator that our hypothesis can be confirmed and the two proposed domains provide valuable use cases for transfer learning. We introduced some potential application scenarios for transfer learning and discussed why we believe the introduced domains are feasible candidates for transferring workflows.

References

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

[2] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.

[3] M. E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: A survey, Journal of Machine Learning Research 10 (2009) 1633–1685.
[4] D. Kudenko, Special issue on transfer learning, KI - Künstliche Intelligenz 28 (2014) 5–6. doi:10.1007/s13218-013-0289-5.
[5] M. Oquab, L. Bottou, I. Laptev, 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, 2014, pp. 1717–1724.
[6] C. Matt, T. Hess, A. Benlian, Digital transformation strategies, Business & Information Systems Engineering 57 (2015) 339–343.
[7] M. M. Richter, R. O. Weber, Case-based reasoning, Springer, 2016.
[8] M. Minor, R. Bergmann, J.-M. Müller, A. Spät, On the Transferability of Process-Oriented Cases, in: A. K. Goel, M. B. Díaz-Agudo, T. Roth-Berghofer (Eds.), 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*, Springer, 2016, pp. 281–294.
[9] M. Minor, M. Herold, Discovery of Analogical Knowledge for the Transfer of Workflow Tasks, in: Artificial Intelligence & Knowledge Engineering (AIKE 2020), IEEE Computer Society Press, accepted for publication.
[10] M. Herold, M. Minor, Ontology-based transfer learning in the airport and warehouse logistics domains, in: Workshop Proceedings of ICCBR 2019, Otzenhausen, Germany, 2019.
[11] M. Herold, 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, 2018, pp. 139–149. URL: http://wi.cs.uni-frankfurt.de/webdav/publications/2018_HeroldMinor.pdf.
[12] B. Falkenhainer, K. D. Forbus, D. Gentner, The structure-mapping engine: Algorithm and examples, Artificial intelligence 41 (1989) 1–63.
[13] T. Könik, P. O'Rorke, D. Shapiro, D. Choi, N. Nejati, P. Langley, Skill transfer through goal-driven representation mapping, Cognitive Systems Research 10 (2009) 270–285.
[14] M. Ragni, G. Strube, Cognitive complexity and analogies in transfer learning, KI-Künstliche Intelligenz 28 (2014) 39–43.
[15] M. Klenk, K. Forbus, Exploiting persistent mappings in cross-domain analogical learning of physical domains, Artificial intelligence 195 (2013) 398–417.
[16] S. Fan, Z. Hua, V. C. Storey, J. L. Zhao, A process ontology based approach to easing semantic ambiguity in business process modeling, Data & Knowledge Engineering 102 (2016) 57–77.
[17] C. Natschläger, Towards a bpmn 2.0 ontology, in: International Workshop on Business Process Modeling Notation, Springer, 2011, pp. 1–15.
[18] M. Rospocher, C. Ghidini, L. Serafini, An ontology for the business process modelling notation., in: FOIS, 2014, pp. 133–146.
[19] A. Richter, Gepäcklogistik auf Flughäfen: Grundlagen, Systeme, Konzepte und Perspektiven, Springer-Verlag, 2012.
[20] SAP, Sap best practices explorer, 2019 (accessed November 29, 2020). URL: https://rapid.sap.com/bp/#/browse/categories/sap_s%254hana/areas/on-premise/packageversions/RDS_EWM_S4.
[21] G. Müller, R. Bergmann, Generalization of workflows in process-oriented case-based reasoning, in: The Twenty-Eighth International Flairs Conference, 2015.
[22] A. Polyvyanyy, S. Smirnov, M. Weske, Business process model abstraction, in: Handbook on Business Process Management 1, Springer, 2015, pp. 147–165.