Towards An Assistive and Pattern Learning-driven Process Modeling Approach

Emanuele Laurenzi, Knut Hinkelmann, Stephan Jüngling, Devid Montecchiari, Charuta Pande and Andreas Martin
FHNW University of Applied Sciences and Arts Northwestern Switzerland
School of Business, Riggenbachstrasse 16, CH-4600 Olten, Switzerland
{emanuele.laurenzi|knut.hinkelmann|stephan.juengling|devid.montecchiari|charuta.pande|andreas.martin}@fhnw.ch

Abstract

The practice of business process modeling not only requires modeling expertise but also significant domain expertise. Bringing the latter into an early stage of modeling contributes to design models that appropriately capture an underlying reality. For this, modeling experts and domain experts need to intensively cooperate, especially when the former are not experienced within the domain they are modeling. This results in a time-consuming and demanding engineering effort. To address this challenge we propose a process modeling approach that assists domain experts in the creation and adaptation of process models. To get an appropriate assistance, the approach is driven by semantic patterns and learning. Semantic patterns are domain-specific and consist of process model fragments (or end-to-end process models), which are continuously learned from feedback from domain as well as process modeling experts. This enables to incorporate good practices of process modeling into the semantic patterns. To this end, both machine-learning and knowledge engineering techniques are employed, which allow the semantic patterns to adapt over time and thus to keep up with the evolution of process modeling in the different business domains.

Introduction

The importance of having good process models, which accurately describe the implemented or intended business processes of a company, is well recognized. Nevertheless, many companies face similar practical problems during the analysis and design phase of their business process management. The domain experts, which are the owners and main users of the business process repository, mostly delegate the analysis, design and construction of business process models to business analysts or to the IT departments, who have the necessary modeling and engineering skills. This leads to the additional effort for communication and collaboration, because modeling experts may lack the necessary expertise from the specific business domain. The domain experts delegate the modeling since they lack of modeling know-how. They neither understand the many BPMN syntax elements nor know how to design correct process models with a good style.

We tackle this problem by proposing an assistive modeling approach that makes use of validated domain-specific semantic process patterns, that are established over time to compensate the lack of business process modeling expertise of domain experts.

The idea of using design patterns as reusable solutions for common design problems could successfully be transferred from the domain of physical construction of cities (Alexander et al., 1977) to the domain of object oriented software design (Gamma, 1995). We reuse this idea for the construction of a semantic repository containing both domain-specific patterns and business process models in the form of an ontology.

Our approach goes beyond the mere syntactical or semantic validation, which is already implemented in current business process modelling tools. With the support of machine learning techniques we aim to learn patterns of good business process modeling that typically reside in modeling and domain expert's heads.

Learning Domain-Specific Semantic Patterns

In software engineering, the adoption of design patterns has shown particular success in assisting programmers to develop software.

In enterprise modeling (which includes the practice of BPM), the use of patterns originates from the field of graph theory addressing the problem of graph pattern matching (Fu, 1995).

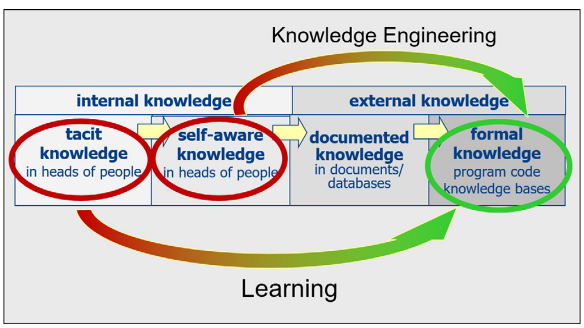
The research in establishing and using patterns to support the process modeling is still quite active (Deng et al., 2017; Delfmann et al., 2010; Gruhn & Laue, 2009). However, these patterns are quite generic and on an abstract level, so that the reuse and application is mainly promoted within the modeling expert community which have the ability to identify the patterns and to apply them in the target domain. In most cases, the domain experts do not have the expertise to adapt the generic patterns to their business domain. Hence, ideal patterns should aim to stay in the background.

This means, a pattern should not only be assistive in terms of (a) syntactic and (b) semantic correctness but also help with appropriate:

c) level of abstraction, which fits the purpose of the process model granularity
d) content (i.e., terminology) fitting to the domain targeted by the process model and
e) modeling styles and conventions, e.g., see scenarios in (Silver, 2011).

In this paper, we define this kind of pattern as a domain-specific semantic pattern for good process modeling.

Several research work evolved around the term *semantic patterns* (Staab et al., 2001; Saif et al., 2014; Soffer et al., 2007) (also known as ontology design patterns (Damjanovic & Violeta, 2009)). However, different from us, they adopt a knowledge engineering approach, which focuses on explicit knowledge that is available from the process models (i.e., syntax and semantic aspects). In our approach, we incorporate a learning approach to deal with tacit knowledge (see left hand side of Figure 1).



We argue that the appropriate *identification* and *construction* of domain-specific semantic patterns can be resolved by combining knowledge engineering approaches with learning approaches. To elaborate on this we refer to the different types of knowledge known from knowledge management that are shown Figure 1.

The identification of an appropriate domain-specific semantic pattern aims to capture the underlying business reality as an appropriate process model.

Based on their experiences, modeling experts—maybe together with domain expert—are able to construct domain-specific semantic patterns. This is regarded as a knowledge engineering task (see upper arrow in Figure 1). However, it would result in a tremendous engineering effort to manually build a sufficient and appropriate set of domain-specific semantic patterns. Furthermore, the process reality to be modeled lays in domain experts' heads but domain experts are not necessary aware of. This means that the knowledge is tacit (Polanyi, 1966). Learning how to identify a domain-specific semantic pattern promises to support this knowledge extraction process (see lower arrow in Figure 1).

The need to learn domain-specific semantic patterns led to formulate the following research question:

- *How to learn domain-specific semantic patterns to support the good process model practice?*

To address this question, we start introducing the three learning sources that are typically adopted as best practices to create a good process model.

Learning Source Cases

We distinguish between three sources of cases from which we can learn domain-specific semantic patterns for good business process modeling:

(a) Modeling expert that give feedback to domain experts' models
(b) Simulation of models
(c) Experience with process executions.

The first learning source typically refers to

(a) syntax
(b) modeling style
(c) level of details of a business process, i.e., abstraction level, and
(d) business process description.

Modeling style and conventions come from experience of experts and are documented, for example, in (Silver, 2011) and Schallert & Rosemann (2012). Syntactical constraints are also documented in the BPMN specifications (OMG, 2011) and implemented in some modeling tools. For example, commercial BPMN tools such as Camunda, Signavio, Bizagi, or Flowable implement syntactical checks that validate syntactical correctness of the design of BPMN models. Both syntax and modeling style can be implemented through the knowledge engineering task. However, learning them would relieve the engineering effort especially for the modeling style and conventions, which sometimes are subjective or application domain-dependent.

Feedback regarding the abstraction level of a business process (c) relies on the intuition and experience of the process modeler, i.e., tacit and self-aware knowledge, respectively. Thus, it can be dealt with a learning approach.

Similarly, feedback regarding the business process description (d) relies on experience of the process modeler. Sometimes, the usable terms within a specific project or application domain are documented, which makes it an alternative source to the feedback. Whereas the former knowledge as implicit but of self-aware type, the latter, which is explicit and belongs to the documented knowledge type.

The second learning source (model simulation) refers to the behavior of process models, also known as behavioral semantics (Muzi et al., 2018 ; Corradini et al., 2017; Mendling, 2009). The most widespread approach in process modeling consists of mapping a process model to a formal semantic like Petri Nets (van Dongen et al., 2008). The approach is used to identify the existence of deadlocks or live-locks through simulation of the correspondent Petri Net model. For example, in the tool BProve (Corradini et al., 2017), Petri Nets are used to simulate the execution of a business process model to assess the safeness and soundness of BPMN collaboration, to check both the existence of deadlocks and proper completion of BPMN models.

Issues like deadlocks or live-locks can be learned by running one of the existing dedicated tools as they sometimes remain difficult to detect even for modeling experts.

The third learning source, (3), refers to the improvement of a process models based on the analysis of the run-time later execution of the business process that can occur manually by the modeling experts or through the support of process mining tools (van der Aalst, 2009).

This list of learning sources reveals that there is still a quite significant amount of implicit knowledge laying in expert's heads, which is neither documented nor implemented in process model tools. By tackling the challenge of explicating such knowledge through learning approaches, we want to show how we intend to address the above introduced research question.

Thus, in the following three sections we introduce three ways of learning a domain-specific semantic pattern: (1) Learning Process Fragment Similarity Model, (2) Learning Pattern's Abstraction Level (3) Learning Pattern's Description. Each of them presents one machine learning approach, which builds on existing techniques.

Learning Process Fragment Similarity Model

Likewise, in case-based reasoning (CBR) a similarity model is an essential component. CBR is known as a technically independent methodology (Watson, 1999) for humans and information systems to reason by remembering (Leake, 1996). During remembering previous cases will be compared by applying a similarity model. The ultimate goal in CBR is then, to transfer or adapt the knowledge from previous cases to the current situation/problem. CBR has its roots in cognitive science, machine learning and knowledge-based systems (Martin & Hinkelmann, 2018). The appropriate engineering of a similarity configuration is a critical requirement for applying CBR.

Martin (2016) introduced an approach, called ICEBERG, how ontology-based case-based reasoning (OBCBR) can be applied in process execution by comparing cases of process fragments. Martin (2016) pointed out that the engineering of the similarity configuration is a critical step and allocates significant resources from domain experts. Martin (2016) and Martin & Hinkelmann (2018) introduced a procedure model for the design and implementation of an OBCBR. Certain procedural stages are essential in this context of configuring a process pattern similarity model as well:

1. At a first stage, it is essential that domain experts decide on an enterprise-specific conceptualization or nomenclature to build a process fragment *characterization* vocabulary or feature set.
2. Then the various *mental similarity* and adaptation models need to be elicited and externalized from the domain experts.
3. Next definition of the *process fragment characterization* will be implemented within an enterprise, domain and/or modeling ontology.
4. Finally, knowledge and domain experts configure the *similarity model* as introduced by Martin et al. (2017) within the ontology. This configuration is made by determining *global* and *local similarity functions* and assigning *weights*.

By describing this approach we applied knowledge-based and knowledge engineering methods in combination and simultaneously. However, there are some drawbacks. The first one concerns the high effort for engineering the characterization, which can be defused by establishing a semantic repository. Secondly, and more difficult to tackle, humans are not good in selecting appropriate similarity features and have difficulties in estimating similarity weights.

A possible way to overcome the mentioned difficulties in selecting similarity features and weights could be taken by prior execution of a data-driven machine learning approach. In one of our previous works (von Rohr et al., 2018) we could show that an initial data-driven machine learning approach based on a regression model can be used to generate, respectively derive similarity features and the corresponding weights.

This section shows how a similarity model can be engineered, how it can be derived and embedded into a knowledge-based environment, and finally how such a model can be learned by applying a data-driven machine learning approach. The resulting similarity model allows retrieving the most similar process model to the one being designed. Later, feedback of domain experts could then re-initialize the data-driven machine learning procedure to determine the adaptation of the similarity measure over time.

Learning Pattern's Abstraction Level

By using the similarity model, it is possible to learn the abstraction of a pattern found in several similar process fragments. As an example, let us consider the process fragment in Figure 2. It models the chain of actions involved in sending an invoice.

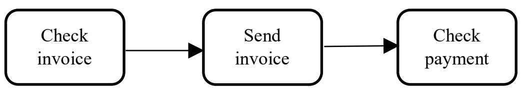


Figure 2: Pattern to send Invoice

This is a semantic sequence that can be observed in almost all the processes that involve sending invoices. Such a pattern can be considered as an abstraction that can be extended in a similar process model being designed as shown in Figure 3.

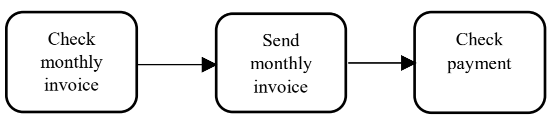


Figure 3: Extension of the Send Invoice pattern

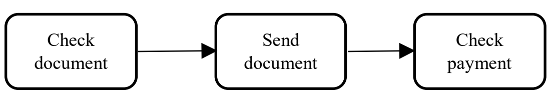


Figure 4: Abstraction of the Send Invoice pattern

It is also possible that the similarity model learns different levels of abstraction for the same pattern over a period of time, as illustrated in Figure 4. Whether the learned abstraction of the pattern is useful or not can be verified through the feedback from the domain experts. The model in Figure 4 shows an example of an abstraction that cannot be used, as sending every type of document does not lead to a payment. Thus, there is a need for a mechanism to incorporate the positive or negative feedback into a pattern repository. CBR has limitations in this respect, as it will create a new pattern based on the feedback but not adapt the level of abstraction of the same pattern. We explore how CBR can be augmented with Reinforcement Learning (Pack Kaelbling et al., 1996) as a mechanism to incorporate the domain experts' feedback – a useful level of pattern abstraction will achieve a positive reward, and an incorrect abstraction will achieve a negative reward or a penalty. Based on the rewards or penalties, the similarity model will be able to identify the right level of abstraction for a process model being designed.

Learning Pattern's Description

In a related work, Wasser & Lincoln (2012) proposed a Process Descriptor Catalog (PDC) to describe activities within a business process according to two main elements, namely Object and Action, and four taxonomies, namely an Action Hierarchy Model (AHM), an Object Hierarchy Model (OHM), an Action Sequence Model (ASM) and an Object Lifecycle Model (OLM). These models were used to organize a set of virtual activities within a particular process domain to express the relationships between actions and object both hierarchically and in term of execution order. Given as input a business process, a *Natural Language Processing (NLP)* system was used to derive the actions and objects involved in the process. By mapping this information to the models, it was possible to assess the similarity between the given process and a possible set of semantically correlated new activities. Then, the choice of the modeler was used as feedback to learn to provide better suggestion adapting the similarity function called in the study, *distance function*.

Extending the approach of activity descriptors to the description of fragments of processes would fit well with the intention of learning pattern's description.

Future Work and Conclusion

In our research roadmap, we intend to implement the three presented machine learning approaches in the form of functionalities of our Agile and Ontology-Aided Modeling Environment (AOAME) (Laurenzi at al., 2018) (see Figure 5). The latter builds on the ontology-based meta-modeling concept described in (Hinkelmann et al., 2018) and seamlessly integrates models with ontologies. Therefore, techniques for semantic annotations/lifting or transformations are not needed as the semantic repository (i.e., see right-hand side of Figure 5) is built while modeling takes place.

The semantic repository contains ontologies reflecting both business process models (i.e., *Model Ontology*) designed by the domain experts and the learned domain-specific semantic patterns, i.e., *Pattern Ontology*. The latter contains either an entire process model or fragments of process models with specific modeling style, conventions, behavioral aspects, abstraction level, etc. It is out of scope of this paper to precisely define what such a pattern consists of.

The learning for the good business process modeling comes from the feedback of a variety of different sources. First, the system can learn and adapt based on the rating of

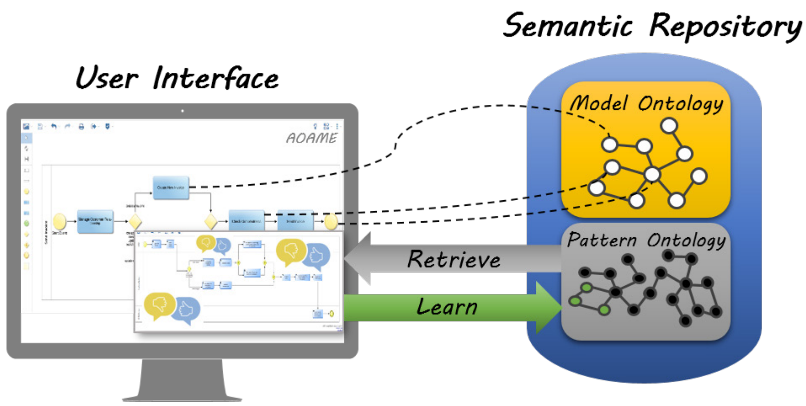


Figure 5. The Assistive and Pattern Learning-driven Approach. Adapted from (Laurenzi et al., 2018)

the modeling experts that give feedback to the models being designed by the domain experts. Second, the process models can be simulated during design time, and the result of the simulation can provide fast feedback to the domain experts while modeling. Third, the domain experts can manually give feedback based on their experience of later process implementation, and the modeling experts can give feedback based on the results from process mining. All of these sources are used to learn best practices and continuously improve the *Pattern Ontology*.

The semantic repository has the benefit of having process models and patterns defined in a machine interpretable representation. Thus, reasoning services and semantic rules can be applied to deduce new knowledge. This was already proven to be successful in several research works in different domains, e.g., Business-IT alignment in the Cloud (Kritikos et al., 2018; Hinkelmann et al., 2016), workplace learning (Emmenegger et al., 2017), and supply chain risk management (Emmenegger et al., 2013).

References

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language : towns, buildings, construction*. Oxford University Press.

Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F., & Vandin, A. (2017). BProVe: A formal verification framework for business process models. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 217–228). IEEE. https://doi.org/10.1109/ASE.2017.8115635

Damjanovic, V., & Violeta. (2009). Ontology design patterns for the semantic business processes. In *Proceedings of the 4th International Workshop on Semantic Business Process Management - SBPM '09* (pp. 51–54). New York, New York, USA: ACM Press. https://doi.org/10.1145/1944968.1944977

Delfmann, P., Herwig, S., Lis, Ł., Stein, A., Tent, K., & Becker, J. (2010). Pattern Specification and Matching in Conceptual Models - A Generic Approach Based on Set Operations. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, *5*(3), 24–43. https://doi.org/10.18417/emisa.5.3.2

Deng, S., Wang, D., Li, Y., Cao, B., Yin, J., Wu, Z., & Zhou, M. (2017). A Recommendation System to Facilitate Business Process Modeling. *IEEE Transactions on Cybernetics*, *47*(6), 1380–1394. https://doi.org/10.1109/TCYB.2016.2545688

Emmenegger, S., Hinkelmann, K., Laurenzi, E., Martin, A., Thönssen, B., Witschel, H. F., & Zhang, C. (2017). An Ontology-Based and Case-Based Reasoning Supported Workplace Learning Approach. In *Communications in Computer and Information Science* (pp. 333–354). Springer, Cham. https://doi.org/10.1007/978-3-319-66302-9_17

Emmenegger, S., Hinkelmann, K., Laurenzi, E., & Thönssen, B. (2013). *Towards a Procedure for Assessing Supply Chain Risks Using Semantic Technologies*. *Communications in Computer and Information Science* (Vol. 415). https://doi.org/10.1007/978-3-642-54105-6_26

Fu, J. (1995). Pattern matching in directed graphs. In Galil Z. & Ukkonen E. (Eds.), *Combinatorial Pattern Matching* (pp. 64–77). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-60044-2_35

Gamma, E. (1995). *Design patterns : elements of reusable object-oriented software*. Addison-Wesley.

Gruhn, V., & Laue, R. (2009). Reducing the cognitive complexity of business process models. In *2009 8th IEEE International Conference on Cognitive Informatics* (pp. 339–345). IEEE. https://doi.org/10.1109/COGINF.2009.5250717

Hinkelmann, K., Laurenzi, E., Lammel, B., Kurjakovic, S., & Woitsch, R. (2016). A Semantically-Enhanced Modelling Environment for Business Process as a Service. In *2016 4th International Conference on Enterprise Systems (ES)* (pp. 143–152). IEEE. https://doi.org/10.1109/ES.2016.25

Hinkelmann, K., Laurenzi, E., Martin, A., & Thönssen, B. (2018). Ontology-Based Metamodeling. In Dornberger R. (Ed.), *Business Information Systems and Technology 4.0. Studies in Systems, Decision and Control.* (pp. 177–194). Springer, Cham. https://doi.org/10.1007/978-3-319-74322-6_12

Kritikos, K., Laurenzi, E., & Hinkelmann, K. (2018). Towards Business-to-IT Alignment in the Cloud. In Z. Mann & V. Stolz (Eds.), *Advances in Service-Oriented and Cloud Computing. ESOCC 2017. Communications in Computer and Information Science* (pp. 35–52). Springer, Cham. https://doi.org/10.1007/978-3-319-79090-9_3

Laurenzi, E., Hinkelmann, K., & van der Merwe, A. (2018). An Agile and Ontology-Aided Modeling Environment. In R.

Buchmann, D. Karagiannis, & M. Kirikova (Eds.), *The Practice of Enterprise Modeling. PoEM 2018.* (pp. 221–237). Vienna: Springer, Cham. https://doi.org/10.1007/978-3-030-02302-7_14

Leake, D. B. (1996). CBR in Context: The Present and Future. In D. B. Leake (Ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions* (pp. 1–35). Menlo Park: AAAI Press/MIT Press.

Martin, A. (2016). *A combined Case-based Reasoning and Process Execution Approach for Knowledge-Intensive Work*. University of South Africa.

Martin, A., Emmenegger, S., Hinkelmann, K., & Thönssen, B. (2017). A viewpoint-based case-based reasoning approach utilising an enterprise architecture ontology for experience management. *Enterprise Information Systems*, *11*(4), 551–575. https://doi.org/10.1080/17517575.2016.1161239

Martin, A., & Hinkelmann, K. (2018). Case-Based Reasoning for Process Experience (pp. 47–63). https://doi.org/10.1007/978-3-319-74322-6_4

Mendling, J. (2009). Empirical Studies in Process Model Verification. In van der A. W. M. P. Jensen K. (Ed.), *Transactions on Petri Nets and Other Models of Concurrency II* (pp. 208–224). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-00899-3_12

Muzi, C., Pufahl, L., Rossi, L., Weske, M., & Tiezzi, F. (2018). Formalising BPMN Service Interaction Patterns (pp. 3–20). https://doi.org/10.1007/978-3-030-02302-7_1

OMG. (2011). Business Process Model and Notation (BPMN), Version 2.0. Object Management Group OMG.

Pack Kaelbling, L., Littman, M. L., Moore, A. W., & Hall, S. (1996). *Reinforcement Learning: A Survey*. *Journal of Artiicial Intelligence Research* (Vol. 4).

Polanyi, M. (1966). *The Tacit Dimension* (Edition 2009 with a new forward by Amartya Sen). Chicago and London: The University of Chicago Press.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2014). Semantic Patterns for Sentiment Analysis of Twitter. In *The Semantic Web – ISWC 2014* (pp. 324–340). Springer, Cham. https://doi.org/10.1007/978-3-319-11915-1_21

Schallert, M., & Rosemann, M. (2012). Prozessreorganisation bei einer Agentur für Unternehmens-, Finanz- und Ressourcenplanung. In *Prozessmanagement* (pp. 579–597). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33844-1_19

Silver, B. (2011). *BPMN Method and Style, Second Edition* (Second Edi). Aptos, CA: Cody-Cassidy Press.

Soffer, P., Wand, Y., & Kaner, M. (2007). Semantic Analysis of Flow Patterns in Business Process Modeling. In *Business Process Management* (pp. 400–407). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75183-0_29

Staab, S., Erdmann, M., & Maedche, A. (2001). Engineering Ontologies using Semantic Patterns. In *IJCAI-01 Workshop on Ontologies and Information Sharing* . Seattle, USA.

van der Aalst, W. M. P. (2009). Process-Aware Information Systems: Lessons to Be Learned from Process Mining (pp. 1–26). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-00899-3_1

van Dongen, B., Dijkman, R., & Mendling, J. (2008). Measuring Similarity between Business Process Models. In *Proceedings of the 20th international conference on Advanced Information Systems Engineering* (pp. 450–464). Springer-Verlag. https://doi.org/10.1007/978-3-540-69534-9_34

von Rohr, C. R., Witschel, H. F., & Martin, A. (2018). Training and Re-using Human Experience: A Recommender for More Accurate Cost Estimates in Project Planning. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (pp. 52–62). SCITEPRESS - Science and Technology Publications. https://doi.org/10.5220/0006893200520062

Wasser, A., & Lincoln, M. (2012). Semantic Machine Learning for Business Process Content Generation. In Meersman R. et al. (Ed.), *On the Move to Meaningful Internet Systems: OTM 2012* (pp. 74–91). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33606-5_6

Watson, I. (1999). Case-based reasoning is a methodology not a technology. *Knowledge-Based Systems*, *12*(5–6), 303–308. https://doi.org/10.1016/S0950-7051(99)00020-9