Statistics-Based Graphical Modeling Support for Ontologies

Andreas Emrich
Institute for Information Systems (IWi)
German Research Center for AI (DFKI)
Saarbrücken, Germany
andreas.emrich@dfki.de

Frieder Ganz
CCSR
The University of Surrey
Guildford, Surrey, UK
f.ganz@surrey.ac.uk

Dirk Werth, Peter Loos
Institute for Information Systems (IWi)
German Research Center for AI (DFKI)
Saarbrücken, Germany
{dirk.werth,peter.loos}@dfki.de

***Abstract*—Models are a conceptualization and aperture of the real world. The asynchronous characteristics of this model construction poses a significant problem especially in highly dynamic and evolving environments. Hence, models need to be permanently checked against the data they represent. From this new challenges for modeling tools arise: Contemporary modeling tools must be able to anticipate environmental events and changes and to provide appropriate support for knowledge engineers. This paper presents a conceptual approach to process events, collect usage statistics and leverage this information for automatic and semi-automatic modeling support of ontologies. In a prototypical evaluation, a plugin for Protegé is developed, that allows for editing ontologies and visualizing new insights according to captured statistics. The paper concludes with two distinct show cases for business process modelling and sensor networks.**

Keywords*—*user modeling support; ontology modeling; ontology evolution; usage evaluation

I. Introduction

Conceptual models formally describe the real world to foster sensemaking and communication [1]. The process of model description takes a certain amount of time. The same applies for the time period of model usage. Consequently, models have to be checked against their real world equivalents permanently [2]. Especially in highly dynamic environments, the pace of environmental changes can be overwhelming for modelers [3], i.e., they need tool support to aid them with anticipating these environmental changes [4].

For the semantic web, notions such as ontology learning, ontology evolution, etc. paint a vision of self-adapting, knowledge-based model bases, which comprehend, reflect and adapt towards their environment. A core challenge in this respect is, that many modeling decisions cannot be decided automatically. For instance, if we capture interactions for a swim guide application for the iPhone, and we get a high correlation on ”good swimming experience” and ”good weather”, but also on ”good swimming experience” and ”bad weather”, it is hard to decide automatically, if the weather does matter or not. The specific domain knowledge of the knowledge engineer could help to resolve this problem. This example demonstrates, how data acquisition methods and human modeling activities are co-dependent from each other. Hence, an approach is needed that unifies these divergent perspectives.

In the context of this paper, we present a conceptual approach for a statistics-driven modeling support that presents an application programming interface (API) for capturing context events and transferring them to our ontology base and for leveraging this data .The paper demonstrates a novel concept, that will be constructed according to a design-oriented research methodology [5]. In a prototypical implementation, we show a prototype plugin for Protegé that supports statistics-enabled introduction of new concepts and properties to an existing ontology. The prototype features support for ontology classes but not yet for instances. In the show case section we discuss the benefits and potentials of this approach in the fields of business process modelling and sensor networks. Overall, this should serve as a starting point for future research to adopt the depicted concept and to improve modeling practice in various domains.

II. Related Work

Over the years, many different editors for OWL-based ontologies have emerged. While many support a tree-based or form-based editing of ontologies and quite a lot of them also support visualization techniques, only a few support graphical editing of the ontologies.

The approach of Dimitrova et al. [6] provides a basic assistance for modeling ontologies based on defined linguistic rules that have been applied to English text. Populous [7] uses a pattern-based approach to transform table-based data into ontology content. Tools like Cicero [8] or Collaborative Protegé [9] focus on non-automatic, collaborative support by creating workplace for collaborative ontology construction. Other tools such as GrOWL [10] or Protegé plugins such as OntoGraf [11], OntoViz [12], OWLViz [13], etc. merely support a declarative graphical editing of ontologies.

Overall, none of the described approaches supports a direct feedback loop from actually monitored usage data with the ontology models at hand. Although, process modeling and sensor data modeling have their specifics, the same shortcomings can be observed in these areas:

1) Process-specific support: The research field of process mining aims at constructing process models from mining process event logs automatically. Although, there has been extensive work in literature and industrial practice regarding this topic, until now there are only a few approaches that reflect the representational bias of process mining [14] or support semi-automatic approaches to use the mined knowledge to construct models manually. A further problem of contemporary approaches is the static mining process itself and the lack of the consideration of interaction. An exception in this area is the work of Hammori et al. [15], in which the authors create a permanent loop between monitored interactions and the modeling tool.

2) Sensor-specific support: In the domain of semantic sensor networks, ontologies are usually defined by ontology engineers only. In our previous work [16] we provided an automatic static creation of a semantic representation of sensor networks. Sensors publish their meta information such as sensor capabilities, energy status and neighbouring nodes to a centralised entity (i.e data sink, gateway) where the information is linked and stored in an ontology. However this approach does not contemplate the higher meanings of the data and does not provide a perceptual view of the sensors environment but only about the sensor devices itself . Recently some novel approaches try to combine machine learning methods to either bootstrap or refine ontologies and represent the meaning of the raw data in a semantic representation.

In the work of Stocker et al. [17] a system is introduced to detect and classify different types of road vehicles passing a street with the help of vibration sensors and machine learning algorithms. The objectives of the work are to acquire knowledge, represented in an ontology by abstracting from the physical sensor layer and the sensor data layer via classification methods.

The outcome of the classification process is then transferred into an ontology representation. The authors use rule based inference to map the outcome of the classifier to the ontology. The ontology consists of concepts such as feature of interest (vehicle type) and observation result time. For each classified car, an individual is created in the ontology with the particular context information.

In the work of Barnaghi et al. [18] abductive reasoning is used to analyse raw sensor data and eventually infer through ruling out obsolete explanations what ontological concept the data refers to.

However, none of the approaches use a supportive semi-automatic approach in which engineers and intelligent algorithms can complement each other. In this paper, we introduce an hybrid approach that on the one hand works autonomously but also supports the decision making process for domain specialists.

III. CONCEPT

The goal of this research, is to provide a tool to support ontology modelling and management by incorporating live data originating from ontology usage. Based on the analysis of related approaches in literatures (cf. Section II), some key requirements can be formulated, that aid to design such a system:

1) **Various event notification modes**: Based on different modelling needs, also different interaction modes need to be supported:
   - *Batch operation*: Historical Events from log files or databases can be populated and taken over into the tool at once.
   - *Real Time interaction*: Every single event that is monitored in real time will be pushed directly to the tool.
2) **Different degrees of model adaptation**: Based on the criticality, importance of certain artefacts or the statistical significance of their correlation, different degrees of adaptation support will be provided by the tool:
   - *Automatic adaptation*: In case statistical significance of an unknown relationship is present, such relationships will be explicitly modelled as properties interlinking the two respective concepts with the relationship.
   - *Semi-automatic adaptation*: Statistical significance of a new relationship or concept is only partly given, this will be modeled and highlighted to be revised manually by a domain expert.

A. Overview of the Approach

The approach chosen aims at being generic, in order to cater the needs of different use cases that expose a strong need for dynamic adaptation and the demand for a deep analysis of interactions with the models. Figure 1 depicts the influences relevant for the considerations made in this approach:

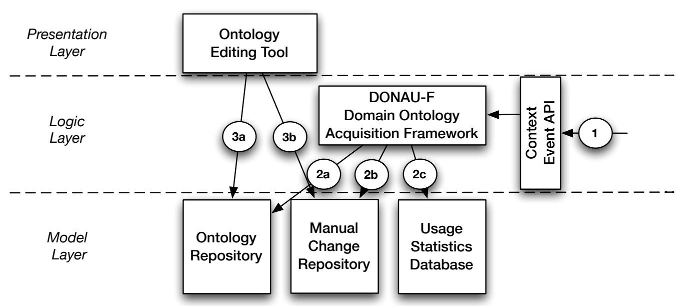

Fig. 1. Ecosystem for Statistics-Driven Ontology Modeling Support

The figure explains the general infrastructure, we provide for ontology evolution support.

1) *Event notification*: A generic context API serves as an entry point for any kind of event that needs to be associated with model data (case 1). A shallow data representation for exchange of such event notification is chosen. In technical terms, this API provides RESTful Web Service interfaces in order to be easily accessed through many different applications and programming languages.

2) *Change reasoning*: The central component for all ontology evolution processing and reading and writing the usage statistics database is the DONAU-F or DONAU framework (Domain Ontology Acquisition Framework) component. It loads the underlying ontologies and infers on the facts stated there. The DONAU-F component provides a plugin infrastructure for data-specific extensions of reasoning mechanisms. See Section V for some examples related to business process modelling

and sensor networks. In terms of the quality and significance of statistical correlations, the changes are either propagated to the ontology base automatically (case 2a) or are marked for later manual editing in the ”manual change repository” (case 2b). In any case, all event statistics are synchronized with the usage statistics database (case 2c), which serves as data repository for all further adaptations taken in the future by DONAU-F.

3) *Ontology editing*: The ontology editing tools load the ontologies in their usual manner, but also highlights relationships and concepts that have been inferred based on our component. It distinguishes between automatically inferred concepts and relationships and such that require a specific modeling action by the knowledge engineer. Section IV describes, how this is implemented in a prototypical version for Protegé.

Overall, every system undergoes changes, that are mainly influenced by the actual interactions with these systems. All these interactions should be covered in a consistent manner, in order to analyse them and derive possible adaptations for the ontology design. The main idea is to correlate observed behaviour, identify artefacts and discover relationships amongst them. According to the statistical significance of these relationships, automatic or manual adaptation plans can be triggered. The core monitoring analyser components fulfils this task, feeds the associated statistics database and eventually causes the ontology to change on its own, or marks certain discovered relationships for later manual editing.

B. Definition of Monitored Aspects

As the main principle has been described, it is still unclear, how such monitoring can be implemented. For a data definition the following aspects have to be considered:

- *defined concepts*: In order to monitor behaviour, it is essential to name aspects clearly, that are monitored, in order to deliver starting point for further analysis. E.g. in our swim app example: If the concepts weather and swimming are already defined, it is easy to analyze a relationship amongst them.
- *wildcards*: For unknown aspects, it is essential to name them according to some variable name, in order to correlate them later on and to name them.
- *defined relationships*: In order to capture the nature of dependencies, it is important to have named relationships that can be either validated or invalidated through the observed behaviour.
- *open relationship interlinking*: The possibility of linking aspects arbitrarily must not be impeded by a superponed model. This is important to ensure that unstructured scenarios are possible as well.

IV. Prototypical Implementation

The described concept has been implemented as a prototype for the ontology modeling software Protegé. To bring about the described changes, the OWLViz [13] plugin has been extended by a new view that supports statistics-enabled, graphical editing of ontologies. The view consists of a graphical editing panel that displays discovered concepts and relationships.

The screenshot in Figure 2 visualizes, how the plugin works. In the graphical editing panel, ontology classes are represented by ovals and properties, that link them are represented with connecting lines. Newly discovered classes and properties are displayed either in green or in red. Green indicates, that the given concept or relationship was significant enough for automatic detection, whereas red indicates that the co-occurrences in the event logs hint at a possible concept or relationship, but because of the low significance it could not be confirmed. The size of the ovals and the thickness of the connecting lines is associated with the relative importance gathered from the underlying statistics, i.e., if a newly discovered class A is used more often than another class B, A is being displayed bigger than B. In our example, the concepts ”Temperature” and ”Light” are new. As ”Temperature” occurs more often in the associated statistics, the term is represented with a bigger oval. In the given example, there are no discovered relationships between these discovered concepts and already established concepts that are significant enough to be displayed, i.e., in order to be displayed, the statistical significance of such relationships has to surpass a defined threshold. In our example, this is 0.3, i.e., concepts have to cooccur in more than 30% of the cases in order to be displayed. The same principle is applied to properties: The stronger a correlation among two classes through a linking property is, the thicker the line is that represents his property. In the given example, there is obviously a stronger correlation between ”Swimming” and ”Good Weather” than between ”Hiking” and ”Good Weather”. Apparently, the relationship between ”Swimming” and ”Good Weather” is significant enough, to infer an automatically discovered relationship.

What cannot be seen in the screenshot is, that statistics are also maintained for already existing concepts and relationships. Although there is no dedicated view for that in our current version of the prototype, a future ”Ontology Management View” should enable to reassess the importance and relevance of certain concepts and relationships. By and large, this can help to ensure that an ontology retains a certain size and thus helping to reduce computation time for querying the ontology base.

A. System Model

The main design goal for our solution is, that it provides an open infrastructure for ontology evolution support, which is independent from the underlying ontologies and the programming language of the source systems. Figure 1 shows the main architecture components of our ontology evolution infrastructure.

It is straightforward to monitor and analyse concepts and relationships that are already defined by the user through the Ontology Editing Tool. However, unstructured information collected via the Context Event API require mechanisms to 1) create new labelled concepts and relationships that reflect the work flow of the underlying processes and 2) to validate and or invalidate existing semantic knowledge.

 ISBN: 978-1-61208-293-6

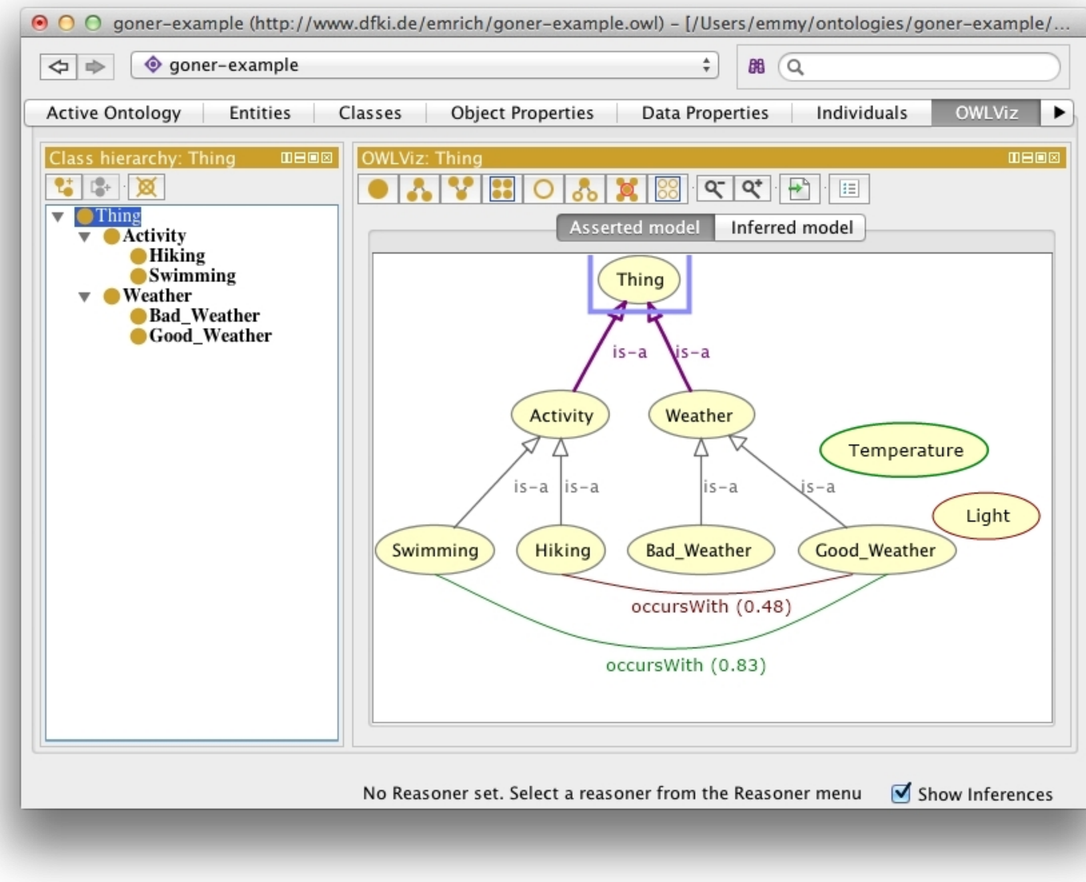

Fig. 2. Graphical Editing View of the DONAU-F plugin

In this section, we discuss DONAU-F, an inference framework to detect wildcard concepts through clustering that groups events similar to their occurrence and context with the help of the k-means clustering algorithm. Furthermore, to detect new linking or invalidate existing linking between concepts, a Markov model is used to create a probability distribution of the temporal relation between different concepts.

1) Wildcard Concept Extraction: In our prototypical implementation, we use a k-means clustering mechanism that groups certain events based on their properties (occurrence, meta information, time) into groups that can either lead to new concepts or be mapped to existing ones. We define a certain threshold that indicates if a new mapping between cluster and concept can be populated without manual revision or if an ontology engineer has to be considered and the new concept therefore has to be highlighted in the editing tool.

2) Open Relationship Interlinking Extraction: Our approach exploits the frequency of events and their temporal occurrence to construct a Markov chain that represents the likelihood of temporal relations and correlation between events. The system counts the occurrence of events and creates a frequency distribution table. The created Markov chain represents the probability of the transition from one event to another event. The chain let us infer if events occur more frequent after certain events and therefore are in some relationship that is going to be represented in the ontology.

Figure 3 shows how temporal relationships are discovered levaring the depicted Markov chain.

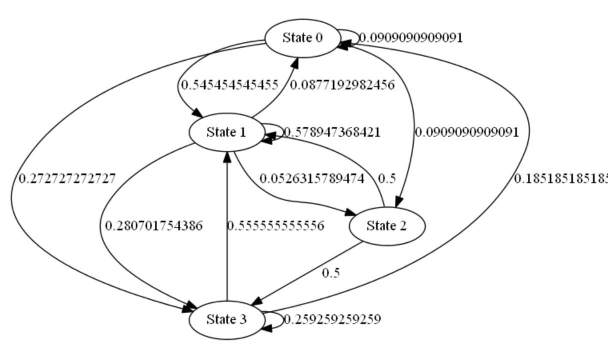

Fig. 3. Relations between different wildcard concepts and their temporal likelihood to occur before/after each other

In our implementation, we are able to detect, represent and highlight relations between concepts through temporal properties such as *occursAfter*, *occursBefore* and *occursSame*. Through hierarchical clustering [19] the system is able to relate concepts through properties such as *isA* and *similarTo*.

 ISBN: 978-1-61208-293-6

V. SHOW CASES

The presented concept and tool support could improve the model management in many different areas. In order to explain the benefits approach, the two areas of business process modeling and sensor networks are explored for application potentials of our approach.

A. Business Process Modeling

For business process modeling the representation of the operational business of a company is a key requirement. However, the business environment is very dynamic and constantly changing. In terms of business process reengineering, according to which the organizational workflows should be reengineered throughout the modeling activities, a big issue is, that the actual workflows and processes change while the modeling efforts are being done. In consequence, once the modeling is finished, it cannot be ensured, that the representation of the actual processes is still correct.

A business process does not only have workflow-related aspects, but also has to consider data, resources, organizational aspects etc. A possible notation for business processes, which covers all these aspects, is the extended event-driven process chain (eEPC). It comprises events, functions, organizational units, resources, documents, etc. as possible design artefacts. The business process modeling notation (BPMN) concentrates on the workflow issues, organizes roles with pools and lanes and introduces ”artifact” for all other aspects.

In the context of business process modeling, our approach could have the following positive impacts:

- *discover process variants*: Especially in processes with many execution alternatives and degrees of freedom concerning execution, it is hard to explicitly model the process without limiting the user. By analyzing process executions in the crowd, process variants can be discovered and the process modeler in charge can easily decide, whether to accept newly emerged process variants as standard operating procedures or to discard them.
- *discover new process steps*: If a process model in terms of an ontology exists, and new unknown process steps are discovered, this might indicate, that the process actually is not compliant to the model, which could have its origin in an insufficient model or an erroneous execution.
- *discover new process responsibilities, resources, etc.*: In the same manner, new responsibilities can be determined. In terms of organizational units this means, that a process can also be executed by other organizational units than originally planned, or it can be identified as a situation, that is not desired. The same applies for associated resources, inputs, outputs of process steps, etc. As many process steps, e.g. a contract checking, rely on deep consistency checks of associated resources and such consistency checks rely on a defined model, it is obvious, that this method helps to ensure that all relevant artefacts are considered by such consistency checks.
- *validate existing relationships*: As mentioned earlier, a defined model does not necessarily reflect the current reality of a system. Therefore, existing model relationships should be constantly monitored and analyzed in order to be validated or invalidated. E.g. if we have have a business process with several subsequent process steps, and according to that process model step B always follows step A, but statistics shows, that in 60% of the cases step C follows step A, the process model needs to be changed accordingly. In terms of evaluation mechanisms this implies, that we need a plugin mechanism as described in Section IV-A to enable a deep semantic analysis of concepts and relationships in an ontology.

Especially in unstructured and not fully modeled scenarios, the approach seems to have its merits and enable an easier maintenance and evolution of business process models. Future research should catch up with process mining and process evolution work in order to achieve the vision described here.

B. Sensor Networks

Sensor Networks are exploited to capture and share data from the physical world and integrate it into software systems. Recently, there has been an increasing trend in research fields such as pervasive and ubiquitous computing and especially in the Smart-Home, -Office domains, where making the gathered data available to the end-user is crucial.

One of the main challenge that remains is to make the usually raw unstructured sensor data understandable for the user and/or machine-interpretable [20]. The Semantic Sensor Web [21] is one approach that allows to annotate, represent and map gathered sensor data to semantic concepts and also represent their relations via properties. Despite the data-centric focus, also device and physical information such as sensor device meta information and hierarchies, environmental attributes and network layouts can be modelled [22].

However the nature of sensor networks, phenomena and data gathered is volatile. Sensor devices can be faulty, parameters and attributes of events can change and other adhoc issues, that alter the association between model and reality can occur. Moreover the vast amount of information produced leads to an information overload that can not be managed by single experts.

The dynamics of sensor networks and observed phenomena has to adapted in the model. We identified several use-cases where the proposed approach can be used to facilitate the adaptation between real state of the network and environment, and the semantic model.

- *Knowledge Acquisition from unstructured raw sensor data*: Events monitored by sensor networks can be modelled by domain experts in the initial ontology as defined items such as ”bad weather” or ”good weather”. However, with the upcoming deluge of data and the information overload for human ontology engineers, this process can be outsourced to the DONAU framework. DONAU can be used to support and facilitate the construction of an initial model or

 ISBN: 978-1-61208-293-6

to refine the representation based on the statistics gathered.

- *Outlier Detection*: The DONAU framework cannot only be used to acquire knowledge that is expected in the domain, but also to infer new insights. Occurring events that cannot be related to existing defined items can be marked as wildcard concepts, and eventually highlighted in the Protege Plugin for further inspection.
- *Network Topology Tracking*: In case that the network topology of sensor networks are modelled in a semantic representation, the approach can be used to monitor changes and update the ontology. The Context API can be used to retrieve health information from particular nodes, in case nodes are not responding or communicating failure, changes can be reflected in the ontology.

VI. Conclusion and Outlook

The synchronization of models and their equivalents represent a core problem of ontology management and information modeling, especially in highly dynamic environments such as the Internet of Things or business information systems. This paper presented a conceptual approach for the capturing of event data within the DONAU framework, that enables to identify automatic changes to an ontology or to provide recommendations for model adaptations to ontology engineers. A prototypical implementation in the ontology modeling tool Protegé demonstrated, how this support functionalities can enhance the graphical modeling of ontologies. Furthermore, we explicated, how the depicted approach can improve the graphical modeling and model management in the domains of business process management and sensor networks.

In terms of evaluation, this paper has shown a first prototypical evaluation as a proof-of-concept. It demonstrates the potential of the depicted approach. However, further evaluations are needed in the future to evaluate the quality of recommendations from an information retrieval (IR) perspective using common IR metrics such as precision or recall and from a user perspective with help by structured user walkthroughs and qualitative questionnaires.

The approach presented in this paper has its focus on a class and not an instance level at the moment. For the consideration of the instance level more aspects have to be considered, as the relationship properties might not be only class-to-class or instance-to-instance but also class-to-instance relationships. Moreover, complex properties are not considered at the moment, that relate to more than two involved artefacts. Clustering methods could guide the way, how to find the most appropriate subsets of artefacts that constitute such relationships and hence are recommended in the tool. Furthermore, aspects as costs or priority as described by Maedche et al. [23] have not been considered so far, but could help to improve future versions. In a similar manner, existing model relationships could be permanently reevaluated regarding their significance. This could help in areas, where relatively compact models are needed, e.g. for high-performance reasoning. The authors plan to release the presented prototypical implementation as an open source software project, in order to provide a tool and code base for a new generation of ontology editing tools, that allow for the seamless integration of the modeling world and actual running systems.

Moreover, the concept that has been proposed is not only limited to ontology modeling and editing but also can help to find potential improvements. In terms of tool support, future implementations could transfer these results to more domain-related tools such as business process modeling suites, etc. Furthermore, the gathered statistics could be part of business intelligence applications for model governance. E.g., non-relevant relationships could be dropped from the model in order to speed up associated analysis. Moreover, dependency analysis of certain events as shown in [24] could help to estimate the impacts of proposed modeling changes and could be a vital feedback for ranking mechanisms. Besides modeling, the applied principles could also be transferred to other problem classes, such as the navigation of ontologies. Based on the research presented in [25] new mechanisms could be developed, that offer a relevance-based navigation of ontologies.

References

[1] Y. Wand and R. Weber, “Research Commentary: Information Systems and Conceptual Modeling?A Research Agenda,” *Information Systems Research*, vol. 13, pp. 363–376, 2002.

[2] G. Y. G. Yang, M. B. Dwyer, and G. Rothermel, “Regression model checking,” *2009 IEEE International Conference on Software Maintenance*, pp. 115–124, 2009.

[3] B. A. Rajabi and S. P. Lee, “Modeling and analysis of change management in dynamic business process,” *International Journal of Computer and Electrical Engineering*, vol. 2, pp. 181–189, 2010.

[4] J.-C. B. A. Francois, D. J. Blackwood, and P. W. Jowitt, “Decision mapping : understanding decision making processes,” *Civil Engineering and Environmental Systems*, vol. 19, pp. 187–207, 2002.

[5] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. [Online]. Available: http://www.hec.unil.ch/ypigneur/HCI/articles/hevner04.pdf

[6] V. Dimitrova, R. Denaux, G. Hart, C. Dolbear, I. Holt, and A. G. Cohn, “Involving Domain Experts in Authoring OWL Ontologies,” *Proceedings of the 7th International Semantic Web Conference ISWC2008*, vol. 401, pp. 1–16, 2008. [Online]. Available: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-401/iswc2008pd_submission_33.pdf

[7] S. Jupp, M. Horridge, L. Iannone, J. Klein, S. Owen, J. Schanstra, K. Wolstencroft, and R. Stevens, “Populous: a tool for building OWL ontologies from templates,” *BMC Bioinformatics*, vol. 13, no. Suppl 1, p. S5, 2011. [Online]. Available: http://www.biomedcentral.com/1471-2105/13/S1/S5

[8] K. Dellschaft, H. Engelbrecht, M. Barreto, S. Rutenbeck, and S. Staab, “Cicero: Tracking Design Rationale in Collaborative Ontology Engineering,” *The Semantic Web Research and Applications*, vol. 5021, no. 5021, pp. 782–786, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-68234-9_58

[9] N. F. Noy and T. Tudorache, “Collaborative Ontology Development on the (Semantic) Web,” *Nature Biotechnology*, 2008.

[10] S. Krivov, R. Williams, and F. Villa, “GrOWL: A tool for visualization and editing of OWL ontologies,” *Web Semantics Science Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 54–57, 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1570826807000157

[11] “OntoGraf Project Website.” [Online]. Available: http://protegewiki.stanford.edu/wiki/OntoGraf

[12] “OntoViz Project Website.” [Online]. Available: http://protegewiki.stanford.edu/wiki/OntoViz

[13] “OWLViz Project Website.” [Online]. Available: http://www.co-ode.org/downloads/owlviz/

[14] W. M. P. V. D. Aalst, “On the Representational Bias in Process Mining,” pp. 2–7, 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5990012

[15] M. Hammori, J. Herbst, and N. Kleiner, “Interactive workflow miningrequirements, concepts and implementation,” *Data & Knowledge Engineering*, vol. 56, no. 1, pp. 41–63, 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169023X05000273

[16] F. Ganz, P. Barnaghi, F. Carrez, and K. Moessner, “Context-aware management for sensor networks,” in *Proceedings of the 5th International Conference on Communication System Software and Middleware*. ACM, 2011, p. 6.

[17] M. Stocker, M. Rönkkö, and M. Kolehmainen, “Making sense of sensor data using ontology: A discussion for road vehicle classification,” in *International Congress on Environmental Modelling and Software*. iEMSs, 2012.

[18] P. Barnaghi, F. Ganz, C. Henson, and A. Sheth, “Computing perception from sensor data,” in *Sensors, 2012 IEEE*. IEEE, 2012, pp. 1–4.

[19] A. Maedche and S. Staab, “Ontology learning for the semantic web,” *Intelligent Systems, IEEE*, vol. 16, no. 2, pp. 72–79, 2001.

[20] O. Corcho and R. García-Castro, “Five challenges for the semantic sensor web,” *Semantic Web*, vol. 1, no. 1, pp. 121–125, 2010.

[21] A. Sheth, C. Henson, and S. S. Sahoo, “Semantic sensor web,” *Internet Computing, IEEE*, vol. 12, no. 4, pp. 78–83, 2008.

[22] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, “The ssn ontology of the w3c semantic sensor network incubator group,” *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012.

[23] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz, “Managing Multiple Ontologies and Ontology Evolution in Ontologging,” *Intelligent Information Processing IFIP The International Federation for Information Processing*, vol. 93, pp. 51–63, 2002.

[24] A. Emrich, D. Werth, and P. Loos, “Real-time semantic process change impact analysis,” in *Innovative Unternehmensanwendungen mit In-Memory Data Management 2011. Innovative Unternehmensanwendungen mit In-Memory Data Management (IMDM-11), December 2, Mainz, Germany*, W. Lehner and G. Piller, Eds. Springer, 2011, pp. 91–100.

[25] A. Emrich, A. Chapko, J. Gehlen, F. Ganz, D. Werth, P. Loos, and D. Campus, “Personalized and Context-Aware Visualization and Navigation of Ontologies for Mobile Services,” in *Proceedings of the 12th International Conference on Human-Computer Interaction with Mobile Devices and Services*, Lisbon, 2010.