Challenges of Modeling and Evaluating the Semantics of Technical Content Deployed in Recommendation Systems for Industry 4.0

Jos Lehmann, Michael Shamiyeh and Sven Ziemer
Bauhaus Luftfahrt e.V., Willy-Messerschmitt-Straße 1, Taufkirchen, 82024, Germany

Keywords: Modeling, Evaluation, Deployment of Semantics, Web Ontology Language, Industry 4.0, Aviation.

Abstract: In the context of Industry 4.0 the Smart Factory is enabled by the automation of physical production activities. The automation of intellectual pre-production activities enables what is here dubbed the "Smart Studio". A key-element of the Smart Studio is Semantic Technology. While prototyping an ontology-based recommendation system for technical content about the case-study of the aviation industry, the problem of the readiness level of Semantic Technology became apparent. This led to the formulation of a Semantic Modeling and Tagging Methodology. The evaluation of both prototype and methodology yielded valuable insight about (i) the quantity and quality of semantics needed in the Smart Studio, (ii) the different interaction profiles identified when testing recommendations, (iii) the efficiency and effectiveness of the methods required to achieve semantics of right quantity and quality, (iv) the extent to which an ontology-based recommendation system is feasible and reduces double work for knowledge workers. Based on these results in this paper a position is formulated about the challenges for the viable application of Semantic Technology to technical content in Industry 4.0.

1 INTRODUCTION

Research on Industry 4.0 develops and evaluates software prototypes to assess how and to what extent digital technology can make industrial processes more flexible and efficient (Vogel-Heuser et al., 2017). As discussed in (Lehmann et al., 2017) and in (Lehmann et al., 2018), digital technology can potentially effect two new paradigms: the Smart Factory, an established concept in research on Industry 4.0, and what, by analogy, can be dubbed the "Smart Studio". In the Smart Factory Robotics, Cyber-physical Systems (CPS) and the Internet of Things (IOT) can enable the automatic reconfiguration of Production and Logistics lines, increasing productivity and reducing costs. In the Smart Studio Knowledge Management (KM) and Artificial Intelligence (AI) can reshape pre-production processes, from Conceptual Design to Prototyping, by supporting or automating the intellectual activities that make enterprise knowledge available when relevant – a goal comparable to what pursued in Knowledge-based Engineering and Ontology-based Design (Li and Ramani, 2007).

A key-element of both the Smart Factory and the Smart Studio is Semantic Technology (ST) (Biffl and Sabou, 2016). The use of semantics in the Smart Factory is characterized by a focus on physical activities. Here observations through sensors can be leveraged in order to update and correct derivations about a given state of affairs. The use of semantics in the Smart Studio, on the other hand, cannot rely on sensing. It therefore has to rely to a larger extent on logical inference, reasoning or other forms of association, in order to harmonize and integrate information sources about multidisciplinary subjects (e.g. textual corpora, model data, expert knowledge) and to flesh out implicit meanings and consequences.

At present it is unclear how easily semantic technology, especially its modeling tools, can help establish the new work flow of the Smart Studio. In (Xu et al., 2018) this point is made more generally for many of the technologies that are being applied in research on Industry 4.0 and that lack the required readiness level.

While prototyping an ontology-based recommendation system for technical content about the case-study of the aviation industry, intending to test to what extent this type of system is feasible and whether it reduces double work for knowledge workers, the problem of the readiness level of Semantic Technology became apparent. This led to breaking down this problem into a number of sub-problems (discussed

Lehmann, J., Shamiyeh, M. and Ziemer, S.
Challenges of Modeling and Evaluating the Semantics of Technical Content Deployed in Recommendation Systems for Industry 4.0.
DOI: 10.5220/0008348503590366
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 359-366
ISBN: 978-989-758-382-7

in Section 2 of this paper) about the modeling and evaluation of semantics when deployed in the Smart Studio. This in turn led to the formulation of a Semantic Modeling and Tagging Methodology (Section 3). The evaluation of both prototype and methodology (Section 4) yielded valuable insight about (i) the quantity and quality of semantics needed in the Smart Studio, (ii) the different interaction profiles that may be identified when testing recommendations, (iii) the efficiency and effectiveness of the methods required to achieve semantics of right quantity and quality, (iv) the extent to which an ontology-based recommendation system is feasible and reduces double work for knowledge workers. Based on these results in Section 5 a position is formulated about the challenges for the viable application of Semantic Technology to technical content in Industry 4.0.

2 HYPOTHESES, PROTOTYPE, PROBLEMS

Research on the Smart Studio usually assumes that the automated fostering of enterprise knowledge is both possible and essential to increasing the productivity of knowledge workers. In order to pin down and partly test this assumption research hypotheses RH1 and RH2 were formulated.

RH1. It is technically feasible to implement an ontology-based recommendation system (OBRS) that provides in real time references to legacy data to knowledge workers as they compile new *technical reports*.

RH2. An OBRS increases the knowledge workers' productivity by supporting the reuse of existing knowledge and avoid double work.

Despite existing successful examples of ontology-based recommendation systems in many domains, the emphasis of RH1 is on the ontological integration of technical content. This type of content is highly structured and detailed, making it more challenging to strike the right balance between the abstraction requirements of ontological integration and the level of precision required by the users of technical content.

In order to test RH1 and RH2 the design, implementation and evaluation of the prototype PR1 was undertaken:

PR1. A prototype OBRS that provides in real-time references to legacy technical data to knowledge workers of an aviation firm as they compile in a text editor new technical reports about aircraft components. The prototype is developed around an existing linguistics- and statistics-based recommendation engine, originally designed for recommendations of short non technical text items. The transition to a OBRS for technical content is accomplished by semantic-linguistic modeling and tagging, not by re-engineering the recommendation engine.

PR1's development was based on the generic architecture for multilingual semantic applications supporting enterprise knowledge reuse shown in Figure 1. As reported in (Lehmann et al., 2018) with the same formulation but more detail, the architecture comprises four main phases (the alternating gray vertical bars, read left to right).

Raw Data Acquisition. Textual data, e.g. PDF files of technical reports and related non-textual data, e.g STEP files of component models, are selected for integration.

Semantic Modeling. Raw data are sampled in order to extract domain knowledge and compile ontological structures that enable semantic tagging.

Linguistic-Semantic Integration. Textual data are tagged based on term frequency of terms described in the ontology. The ontology is also used to semi-automatically tag non-textual data. All tagged data are then stored in a database.

User Interaction. The database is dynamically accessed by a text editor to provide end-users with recommendations of existing reports or models that may contain information that is relevant to the contents being typed in.

(Chen and Wu, 2008) discusses an architecture for a comparable class of document recommendation systems, though based to a larger extent on user preferences rather than semantic modeling.

Working on PR1 raised two groups of problems about the scaling-up of established approaches to data integration. On the one hand, data-related problems (DP1 through DP6 below) had to do with the optimization of software components that, as shown in Figure 1, are located in the parts of the architecture, that support the preparation and enrichment of the tens of thousands of documents available to the OBRS plug-in. Although not further discussed in this paper the list of these problems provides some context on issues of data preparation required by an OBRS.

DP1. Large Quantity of Documents.

DP2. Layout of Documents.

DP3. Large Size of Documents.

DP4. Integration of additional and relevant Metadata avalaible to Data Owner.

DP5. Suitable Client-Server Architecture.

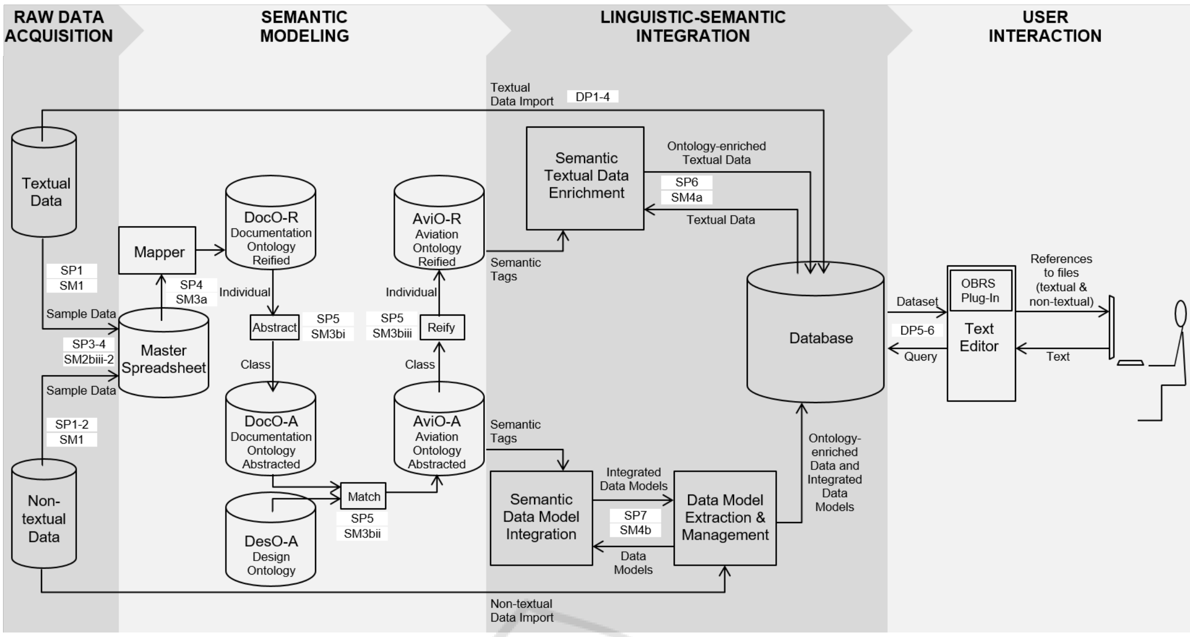


Figure 1: Architecture of multilingual semantic application, OBRS PR1, supporting enterprise knowledge reuse.

DP6. Connection to Data Sources.

On the other hand, semantics-related problems (SP1 through SP7) surfaced in various parts of the architecture shown in Figure 1. They in large part had to do with providing the OBRS plug-in with semantics of the right quality in the right quantity.

SP1. Coverage of the Domain Knowledge required by the Use Cases.

SP2. Representation of the High Level of Detail of the Vocabulary of the Model Data.

SP3. Representation of the Lexical Variations and Multilingualism of the Documents.

SP4. Extraction of Large Quantities of Knowledge.

SP5. Ontology Reusability.

SP6. Semantic Tagging of Textual Data.

SP7. Semantic Tagging of Model Data.

3 SEMANTIC MODELING AND TAGGING

As reported in (Lehmann et al., 2018) the engineering of appropriate ontologies requires to accurately model the information ecosystem in which the OBRS is meant to operate. On the one hand, a normative modeling approach makes sure that documents are correctly classified with respect to the domain of reference. On the other hand, when deploying an ontology onto the corpus, issues specific to the corpus interfere with correct interpretation: lexical variations, multilingualism, abbreviations, technical data.

Figure 2 presents an updated version of the methodology introduced in (Lehmann et al., 2018), which supports the transition of information from sample data to an ontology. After sampling a descriptive document, e.g. a nomenclature for a landing gear extension/retraction system, a conversion into a spreadsheet takes place in step SM2a. Then, the contents (found in headers, lists, indexes, tables etc.) are classified as individuals of a minimal number of classes (e.g., *system*, *component*, *part*) and ordered by an hyponym property (e.g., *narrower-than*). For instance, the individual of class *component retraction actuator* is *narrower-than*, i.e. an aspect of or a functional part of, an individual *landing gear extension/retraction system* of class *system*. In turn such system is *narrower-than*, i.e. an aspect of or a functional part of, individuals *nose landing gear* resp. *main landing gear* of class *component*. Finally, these components are aspects of an individual of class *mastersystem*, which classifies the most generic systems for inferential convenience. Similarly, classes *mastercomponent* and *masterpart* group the most generic components resp. parts.

The resulting hierarchy mixes-up class/subclass, class/instance, whole/part hierarchies, which are disentangled in step SM2(b)iD. After ontological checks and name assignments (SM2(b)i to SM2(b)iii), the OWL ontology (DOCO-R) is set-up via a mapper.

SM1. Sample Data
sample reports and design models that are relevant to use-cases

SM2. Master-Spreadsheet

(a) Prepare Sources
export content of existing multilingual documentation (such as pdf files of reports as textual data, spreadsheets of component hierarchies generated from design models as non-textual data) into a spreadsheet and delete irrelevant parts

(b) Create Ontology's Basic Version

i. Ontological Modeling
first version of ontology is created in reified form

A. Create Classes and Hyponym Property
from sources' section names, table headers

B. Create Individuals
from sources' section content, table entries

C. Assert Hyponym Property between Individuals
from sources' section content, table entries

D. Qualify Hyponym Property between Individuals
disentanglement of relationships class/subclass, class/instance, whole/part

ii. Ontological Checks

A. Translate Individuals IRI's
into main language of ontology

B. Find Duplicate Individuals
as exact match, partial match, no match

iii. Assign names to Individuals as Annotations
including synonyms, abbreviations, their grammatical variations (plural, cases)

SM3. OWL Ontology

(a) Export ontology
from Master Spreadsheet to DocO-R

(b) Integrate ontology

i. Abstract Individuals to Classes
from DocO-R to DocO-A

ii. Match Classes
between DoC-A and DesO-A resulting in AviO-A

iii. Reify Classes to Individuals
from AviO-A to AviO-R

(c) Assess Coverage of Ontology

(d) Merge Ontology

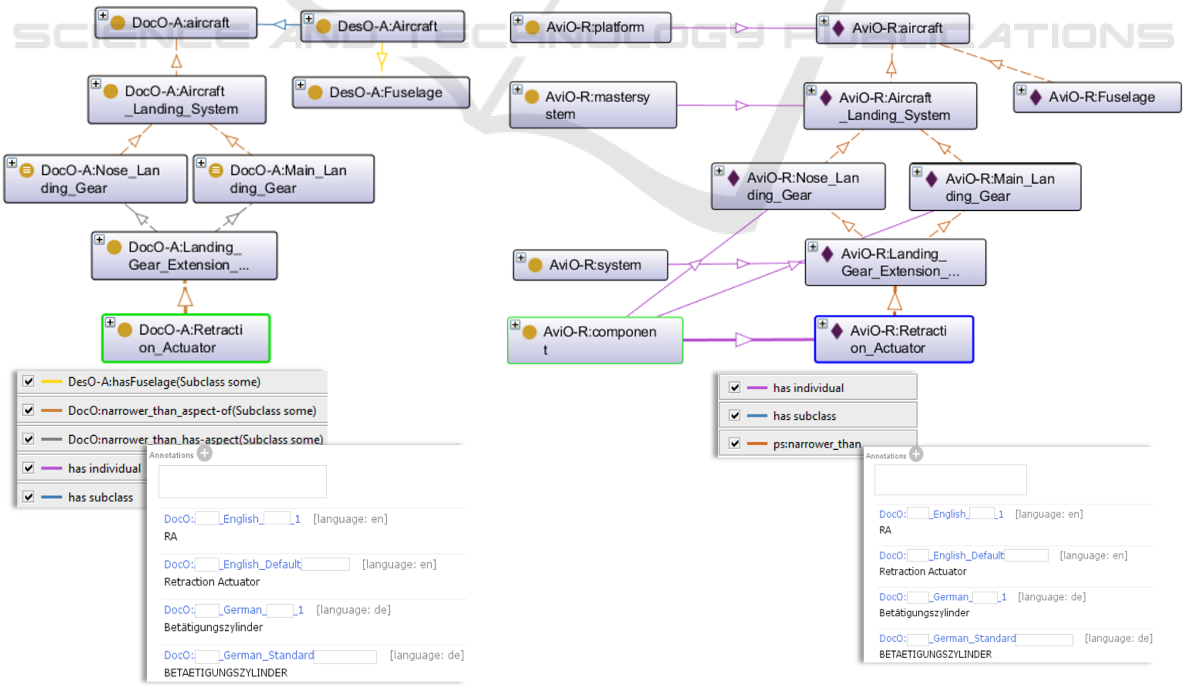
(e) Export Ontology

SM4. Semantic Tagging

(a) Textual Data Enrichment

(b) Data Model Integration

Figure 2: Semantic Modeling and Tagging Methodology.



(a) AviO-A (b) AviO-R

Figure 3: Retraction Actuator in AviO.

DOCO-R is then *abstracted* as DOCO-A via a conversion function, which outputs a representation of the domain knowledge, which exploits the full potential of DESCRIPTION LOGIC (DL) and does away with classes such as *mastersystem*, *mastercomponent*, *masterpart*, introduced in the reified version of the ontology for inferential convenience. In DOCO-A the appropriate parts of the contents of DOCO-R are converted into a DL class hierarchy, while the rest is represented by appropriate object or data properties. DOCO-A can be matched with relevant ontologies, e.g. a Design Ontology DESO-A, to get to a version that is called Aviation Ontology (AVIO-A in Figure 3a). In this example the class *aircraft* of DOCO-A is matched with the class *Aircraft* of DESO-A thereby acquiring a part-of relation to the class *Fuselage* of DESO-A, not found in the original sample data, given their focus on the extension/retraction system under consideration. AVIO-A is then reified as AVIO-R, shown in 3b. AVIO-R reduces the number of properties and classes, reducing the expressivity of the ontology, while retaining all annotations.

Both conversions (Abstract, Reify) are akin to so-called meta-modeling techniques presented for instance in (Welty, 2006), (Glimm et al., 2010), (Jekjantuk et al., 2011).

Steps SM3c through SM3e involve checks of AVIO's coverage with respect to external knowledge sources, as well as the transformation of its import structure and of its serialization, to enable its use by the tagging modules. These steps are not further discussed in this paper.

Finally, AVIO-R is used by the Semantic Textual Data Enrichment module to integrate the term-frequency-based tagging of the documents with extra tags derived from the hyponym hierarchy. AVIO-A is used by the Semantic Data Model Integration module to provide an expert user of the Data Model Extraction & Management module (not shown in Figure 1) with semantic tags for the non-textual data.

4 EVALUATION

The evaluation of PR1 was based on (i) user feedback given through questionnaires, (ii) screen shot sequences of user interactions that included the assessment by users of the relevance of recommendations, (iii) interviews, (iv) developer feedback given through reporting. All this provided evidence on the following issues:

1. How do PR1's quantity of semantics (in terms of domain coverage) or quality of semantics (in terms of level of detail or of lexical variations) score on the following scale?

   [`too poor`, `poor`, `ok`, `rich`, `too rich`]

   Note that values `poor` and `rich` can either have a positive or a negative connotation, therefore `too poor` resp. `too rich` are used to indicate a value beyond what is considered practical by the evaluator.
2. Which types of interaction can be observed between PR1 and its users and how productive are such interactions?
3. How do the methods described in Figure 2 score on the following scales?

   [`partially efficient`, `efficient`]
   [`partially effective`, `effective`]

   These scales are intended for the assessment of the amount of effort needed (efficiency) when applying a given method for the solution of a semantic problem i.e. for modeling the minimal amount of knowledge required by the use cases of a OBRS (effectiveness).
4. Does the evaluation of points 1 through 3 above confirm PR1's feasibility (RH1) and its role in reducing double-work for knowledge workers RH2)?

4.1 Evaluation of Semantics in PR1

As shown in Table 1 the quality (in terms of domain coverage) and the quantity (in terms of level of detail and lexical variations) of the semantics of PR1 was evaluated from the perspectives of three stakeholders: the ontology developer, the plug-in developer, the plug-in user.

The coverage of the domain knowledge required by the use cases was evaluated as `ok` by the ontology developer and the end user but `poor` by the plug-in developer. The ontology developer was satisfied by the amount of domain knowledge provided to PR1 by sampling representative documents. The end user did not notice obviously lacking semantic tags among those the OBRS plug-in provided with any given recommendation – although the end user did not always agree with some tags that were used for a recommendation. The plug-in developer who, as opposed to the plug-in user, had direct access to the ontology, expected more domain knowledge to drive the recommendation engine, beyond what was harvested by sampling the data pool.

The representation of the high level of detail of the vocabulary of the model data was `rich` for the ontology developer, because such data provides a lot of information about class and part-of hierarchies for

Table 1: Evaluation of semantics in PR1.

Semantics Problem	View of Ontology Developer	View of Plug-in Developer	View of Plug-in User
Coverage of Domain Knowledge SP1	ok	poor	ok
Representation of Detail Level of Model Data SP2	rich	too rich	too rich
Representation of Lexical Variations SP3	ok	poor	n/a

components, down to their smallest and most common parts (e.g. O-Ring). From the perspectives of both the plug-in developer and user this level of detail was `too rich`, i.e. problematic, because the list of recommendations could at times be clogged with recommendations that were irrelevant despite containing many occurrences of a term, e.g. O-Ring, which appeared in the input text and was very frequent in the data pool.

Finally, the representation of the lexical variations and multilingualism of the documents varied: for some entities in AVIO lexical variations of their names was evaluated as `rich`, having been modeled extensively, while for other entities the lexical variation was `poor`. This led to some dissatisfaction on the plug-in developer's part again based on the expectation of as much structure as possible to drive the recommendation engine.

4.2 Interaction Profiling

The relevance assessment of recommendations by the users allowed to identify the interaction profiles described below.

Unaimed search-Experimental tester (UE): In this type of interaction the OBRS users saw the system as a means to understanding more about a theme and to finding documents beyond what they already knew or expected as recommendations given the text they input. Users might not always be satisfied with the recommendations – especially regarding the ontological or linguistic categories used by the system to tag a given document. They appreciated though the interaction with a system that helps to explore the data pool.
Frequency: 30%

Aimed search-Prudent tester (AP): In this type of interaction users – similarly to UE interactions – saw the system as a means to exploring the data pool, aiming though at getting a specific document high up in the recommendation list.
Frequency: 10%

Very aimed search-Conservative tester (VC): In this type of interaction users saw the OBRS from the standpoint of their usual way of working. They sought recommendations through a controlled interaction with the system, i.e., they often typed in just the name of a specific report, a part number, a single concept or even only an acronym in order to test whether the system was able to deliver a very specific document they expected. In these interactions users were interested in testing whether the system could perform as well as a user who, based on expertise and knowledge of the data pool, knows exactly what to look for and how to find it.
Frequency: 60%

In general, users who preferred VC interactions evaluated PR1 less favorably than users who preferred UE and AP interactions. In this respect, the increase in productivity yielded by avoiding double work seems to be more readily available to users who are ready to invest time in exploring the data pool through the recommendation list – either because they are interested in exploring new aspects of the data pool to avoid duplication of effort or because they lack expertise and are uncertain which existing documents they need for a given task.

4.3 Evaluation of Methods

As shown in Table 2 the modeling and deployment methods employed for the semantics of PR1 was mainly evaluated from the perspective of the ontology developer.

Sampling textual and non-textual data has proven to be an `efficient` way of achieving the minimal coverage and level of detail required by the use cases. The first two rows of Table 1 suggest though that sampling can only bring so far in terms of a satisfactory behavior of an OBRS. As the number of use cases handled by the system grows, more ontological modeling is needed to semantically consolidate the system.

Assigning names to individuals as annotations for lexical variations is `effective`, as it increases the system's reach on the data pool with limited ontological modeling. As opposed to ontological modeling, annotating the ontology is still time-consuming though and more likely to create redundancies.

Tabulating and mapping for knowledge extraction is very `efficient`, as spreadsheets provide a first cut of the ontology in a format in which it is easier than

Table 2: Evaluation of methods for handling semantics.

Method for Problem	View of Ontology Developer	View of Plugin Developer
Sampling for Coverage and Detail SM1 for SP1, SP2	efficient, partially effective	n/a
Annotating for Lexical Variations SM2(b)iii for SP3	partially efficient, effective	n/a
Tabulating and Mapping for Knowledge Extraction SM2 for SP4	efficient, partially, effective	n/a
Abstraction and Reification for Reusability SM3(b)i SM3(b)iii for SP5	partially efficient, effective	n/a
Textual Data Enrichment for Semantic Tagging SM4a for SP6	efficient, partially effective	efficient, effective
Data Model Integration for Semantic Tagging SM4b for SP7	partially efficient, effective	n/a

in ontology editors to sort and to identify explicit or semi-implicit duplicates in the sample data. Tabulation is not `effective` for more advanced modeling and does not support the logical checks provided by ontology editors.

Abstraction and reification are `effective` ways of achieving reusabilty of semantic modeling results, when starting from sample data and a hyponym-type of relation. On the other hand, implementing and maintaining the abstraction and reification conversion procedures is time-consuming and requires non trivial ontological choices.

The two semantic tagging methods employed on textual resp. non-textual data have complementary efficiency and effectiveness. SM4a can rely on a high degree of automation because it is based on Natural Language Processing and on term frequency, yielding results though that do not always effectively mirror the core semantics of a given document. Conversely, SM4b is driven by user interaction thereby tagging component models more effectively, at the cost of time consuming tagging sessions.

4.4 Evaluation of RH1 and RH2

The overall evaluation of problems DP1 through DP6 (not discussed in this paper) and SP1 though SP7 allowed to establish that PR1 is technically feasible (RH1) and that the PR1 improves the productivity of knowledge workers (RH2). PR1 is most effective in helping knowledge workers to learn about the semantic structure of the data pool as well as about the position of single documents within that structure.

5 CONCLUSION

Work on the development and the evaluation of PR1 has helped identify a number of challenges for the viable application of semantic technology in Industry 4.0.

On the one hand, many of the steps that allow for the preparation and consolidation of semantic structures for a given organization (e.g. the steps described in Figure 2) should be integrated in normal, day-to-day Knowledge Management activities. Organizations that undertake the transition to a paradigm like the Smart Factory or the Smart Studio need to develop the necessary expertise to be able to assess the optimal ratio between effort and semantics quantity and quality when deployed in recommendation or other computing systems. These Knowledge Management activities should be supported by the automation of the following functionalities:

1. Knowledge extraction in a tabular format (e.g. a spreadsheet), rather than in an ontology editor, to make it easier to bootstrap ontologies from sample data by operating (sorting, filtering, matching, editing) on a large amount of raw input information. This is needed in order to make a semantic structure specific to its context of use and enable its deployment in a specific information eco-system. An OWL editor becomes essential at the later stages of ontological analysis, refinement and general management of a consolidated ontology.
2. Meta-modeling, i.e. (class) abstraction and reification to learn and reuse ontologies.
3. Automated matching and automated coverage assessment, to enrich ontologies.
4. Gold standard preparation of a representative subset of the data pool through systematic semantic tagging, to enable the user-independent evaluation of ontology-based systems.
5. Automated monitoring of user-interactions, to achieve a behavioral definition of the values in the scales used in Section 4 (e.g. `too poor`, `poor`, etc.) and of the interaction profiles.

On the other hand, the end-user needs to be supported in the interaction with the recommendations by means of the following functionalities:

6. Explanations of the recommendations with respect to the input text.
7. Migration of OBRS to applications other than text-editors.
8. Configuration of queries relative to user profile or function in the organization.

ACKNOWLEDGMENTS

The project underlying this research (EFFPRO_4.0 – Integration and Analysis of Design and Production Data for a more efficient Development Process Chain) has received funding from the German Federal Ministry for Economic Affairs and Energy under grant agreement no. 20Y1509E.

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

Ontology editing and presentation in Protégé http://protege.stanford.edu, US NIGMS/NIH grant no. GM10331601.

REFERENCES

Biffl, S. and Sabou, M., editors (2016). *Semantic Web Technologies for Intelligent Engineering Applications*. Springer.

Chen, D. and Wu, C. (2008). An ontology-based document recommendation system: Design, implementation, and evaluation. In *Pacific Asia Conference on Information Systems, PACIS 2008, Suzhou, China, July 4-7, 2008*, page 206. AISeL.

Glimm, B., Rudolph, S., and Völker, J. (2010). Integrated metamodeling and diagnosis in OWL 2. In Patel-Schneider, P. F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J. Z., Horrocks, I., and Glimm, B., editors, *International Semantic Web Conference, Revised Selected Papers, Part I*, volume 6496 of *Lecture Notes in Computer Science*, pages 257–272. Springer.

Jekjantuk, N., Pan, J. Z., and Qu, Y. (2011). Diagnosis of software models with multiple levels of abstraction using ontological metamodeling. In *Proceedings of the International Computer Software and Applications Conference*, pages 239–244. IEEE Computer Society.

Lehmann, J., Heußner, A., Shamiyeh, M., and Ziemer, S. (2018). Extracting and modeling knowledge about aviation for multilingual semantic applications in industry 4.0. In *Sixth International Conference on Enterprise Systems, ES 2018, Limassol, Cyprus, October 1-2, 2018*, pages 56–60. IEEE.

Lehmann, J., Shamiyeh, M., and Ziemer, S. (2017). Towards integration and coverage assessment of ontologies for knowledge reuse in the aviation sector. In Fensel, A., Daniele, L., Aroyo, L., de Boer, V., Darányi, S., Elloumi, O., García-Castro, R., Hollink, L., Inel, O., Kuys, G., Maleshkova, M., Merdan, M., Meroño-Peñuela, A., Moser, T., Keppmann, F. L., Kontopoulos, E., Petram, L., Scarrone, E., and Verborgh, R., editors, *Joint Proceedings of SEMANTiCS 2017 Workshops co-located with the 13th International Conference on Semantic Systems (SEMANTiCS 2017), Amsterdam, Netherlands, September 11 and 14, 2017.*, volume 2063 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Li, Z. and Ramani, K. (2007). Ontology-based design information extraction and retrieval. *AI EDAM*, 21(2):137–154.

Vogel-Heuser, B., Bauernhansl, T., and ten Hompel, M., editors (2017). *Handbuch Industrie 4.0 Bd.1-4*. Springer Reference Technik. Springer.

Welty, C. A. (2006). Ontowlclean: Cleaning OWL ontologies with OWL. In Bennett, B. and Fellbaum, C., editors, *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA, November 9-11, 2006*, volume 150 of *Frontiers in Artificial Intelligence and Applications*, pages 347–359. IOS Press.

Xu, L. D., Xu, E. L., and Li, L. (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research*, 56(8):2941–2962.