Modelling of Child Seat Dependencies for Digital Information Services using Semantic Technologies

Sophie Bischoff[1] *

[1] Karlsruhe University of Applied Sciences, Faculty of Information Management and Media, 76133 Karlsruhe, Germany

Abstract. With the increasing importance of knowledge management and the growing amount of data in industry, a field of application for semantic technologies has emerged with the representation of knowledge in the area of technical communication. Semantic networks and ontologies offer a high degree of intelligence and can make complex relationships interpretable for humans and machines alike. This aspect can lead to an improvement in the classification, connection, and delivery possibilities of content. Technical documentation is one of many areas where semantic technologies and knowledge graphs can add value to industry. This paper should give an outlook on how semantic technologies can be used in different use cases by companies in the domain of technical communication in the future. Based on a use case of a German company of the child seat industry, the use of semantic technologies and their various use cases in technical communication is shown. The company has started to use semantic technologies to simplify processes and to provide their digital information products in a user- and context-oriented way. This paper focuses on the possibilities that the use of semantic technologies, especially ontologies, provides in a company. It also highlights the difficulties that can arise in the transition and transformation of data.

1 Introduction

Semantic technologies have been established in technical communication since a certain period for instance in the creation and management of metadata in the field of content management. Recently, semantic technologies have gained considerable influence and attention in other application fields as well. Especially in parts of the industry that deal with complex products or contexts. Semantic networks or ontologies can be used to map these complexities and make them interpretable for humans and machines. The ontologies created can provide solutions for a wide variety of use cases.

Based on the use case from industry, it is shown how ontologies are created and how they can be used. The company is a leading manufacturer of child safety products such as child seats, infant carriers, and prams. The main goal of the company is to build a highly user-friendly knowledge service. The initial problem for creating the Product Knowledge Service arises from the relationship construct of vehicle, product, and child. It is not possible to install every child seat in every vehicle. Installation depends on various structural and legal conditions that must be met. Whether a child seat may be used depends on the individual vehicle seats, not the entire vehicle. In addition, there are different methods of fastening child seats in vehicles and therefore different conditions must be met on the vehicle seats. Hereby it is possible for a product to be approved for using in a car on different seats, but only with different fastening methods. Another regulation for a release is determined by the child to be fastened. In the company's product world, there are products that are approved for different age, height, or weight classes of children. These classes determine with which fastening method a child may be fastened in the child seat. Accordingly, there is an M:N relationship between the vehicle seat and the child seat, depending on the different mounting options on a vehicle seat. There is also an M:N relationship between the child and the child seat, depending on the fastening methods options for a child of a particular age, height, or weight class. These approvals per vehicle seat per product are managed in the type lists and made available to users. The goal is to replace the tabular display of the type list with the Product Knowledge Service.

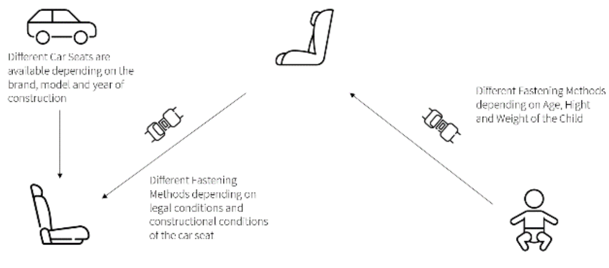

Fig. 1 Dependencies of car seat, child seat and child

The downstream goal of the Product Knowledge Service is to be able to cover further application areas via the ontology created for this purpose. The service is intended to link different applications and provide a complete knowledge service for the child safety area.

* Email: bisol012@h-ka.de

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

This service is to be built based on a content delivery portal (CDP) and additionally contains interactive parts to display directly the necessary content to the user. In addition, the previous processes of content management, creation and updating are to be simplified. For this purpose, an ontology was created that depicts the product world and the interdependencies between them. Using the RDF query language SPARQL, the information is queried, output and displayed in a user interface. The automated linking between individual contents is solved by using Semantic Correlation Rules (SCR). To facilitate the processes, a concept for an administration interface was created. This provides an interface with different functions such as creating, deleting, or modifying data records in the ontology. In the background, SPARQL queries access the ontology and execute the functions. The knowledge service was conceptualised and prototypically implemented in the first stages.

2 Concepts

2.1 Semantic technologies

The use of semantic technologies has become established in technical communication, especially in the sense of the Semantic Web. Information is distinguished and linked to each other through the definition of metadata. Accordingly, knowledge can be represented, retrieved, reused, and made available. The Semantic Web refers to a concept in which data is enriched with structured data. The inventor and founder of the Semantic Web is Tim Berners-Lee, who first defined the concept in 2001 as an extension of the Word Wide Web (WWW). The underlying idea is to enrich content on the web with meta-information so that it is not only comprehensible to humans but can also be interpreted by machines. While humans take further information about a concept from the context and build links unconsciously, this context must first be taught to machines [1]. To relate information and thus make it interpretable, various standards have been established.

2.2 SPARQL

One of the standards of the semantic web is SPARQL. The recursive acronym SPARQL stands for SPARQL Protocol and RDF Query Language. Through standardisation by the World Wide Web Consortium (W3C), the language became a standard for querying RDF-specified information [2ic]. In addition to the query, SPARQL is also responsible for the protocol and presentation of the results. A formal semantics for the query language was defined in the standardisation process. Many features of SPARQL are inspired by SQL, both in terms of syntax and semantics. Individually, the features are intuitive to describe and understand. By combining them, SPARQL can be extended to a complex language. The query language is based on simple queries to RDF in the form of graph patterns. The queries and outputs can be modified via advanced functions such as filter conditions or formatting [3].

SPARQL queries are based on simple subject-predicate-object queries (SPO queries). These queries consist of a triple pattern that corresponds to an RDF triple in terms of structure. The difference is that each position can stand for an information resource or a variable [4].

```
SELECT      ?subject
WHERE
{
      ?subject :predicate ?object .
}
```

Fig 2 SPARQL query triple pattern querying all subjects

2.3 Ontologies

The term ontology is to be understood as equivalent to the term knowledge base. In the simplest sense, an ontology describes a document created in RDF(S) or OWL that models knowledge of an application domain. It is a precise representation of a topic based on consensus. By formalising conceptual structures through the sets of concepts and the relationships between them are represented, machines are supported to interpret content rather than simply represent it. Ontologies represent the most complex form of knowledge representation [4]. Basically, ontologies are divided into two different types. Lightweight ontologies contain instances, taxonomies and relations. This type of ontology is usually used to model ontologies for specific domains. Heavyweight ontologies are an extension of lightweight ontologies. They are enriched with axioms and constraints for a deeper semantic interpretation of concepts and relations.

Instances basically describe the unique occurrence of something. They represent real existing elements of a domain [5]. This means that all objects are referred to entities and include both concrete objects such as persons and abstract objects such as organisations. Classes describe the various concept categories. A class is an exact description of the requirements for the instances that can belong to this class. These are usually organised in ontologies as taxonomies, so that the inheritance mechanisms already considered are applicable. Relations describe dependencies between classes. The relations serve a clear description of the classes and entities represented. In the taxonomy, the " is-a" relation is defined by default. All other relations must be explicitly defined.[6].

2.4 Modelling of ontologies

The modelling of ontologies is a complex task that requires both a good understanding of the domain to be modelled and of the modelling language used. In this work, OWL is assumed to be the modelling language. In contrast to semantic technologies, no standard procedure has yet been established for modelling. The existence of corresponding language standards such as OWL offers a formal framework that can be used as a guide in the creation of ontologies. However, they do not provide an answer to the question of how a good ontology is created [5].

A model always represents a part of the world to some extent, but never completely and considering all aspects. Since a model can be created in different ways, the expected use should be defined at the beginning. With the help of a specification, the intended use and the target users are defined. The first step is to define the area of the application domain to be modelled. A fundamental problem in the creation of ontologies is the complexity of the modelled relationships on the conceptual and formal level. Because of this, a definition of the level of detail is necessary [5].

An important difference in the type of modelling is the level at which relations are assigned. The distinction lies in whether the modelling is instance-based or class-based. In the instance-based type of modelling, relations to other instances are assigned to each instance at the lowest level. The conception of such an ontology has a low complexity, since hardly any abstraction of the data is necessary. On the other hand, the creation, is much more complex since relations on the same level cannot be inherited. The modelling method at class level is far more complex in the conceptual phase since the classification of the objects must have a high degree of accuracy. Otherwise, wrong properties are inherited by the instances and contradictions arise within the ontology [5].

2.5 Semantic correlation rules

SCR can be defined as a general concept with which content can be automatically linked to each other via semantic classification. SCRs can be used to map logical content correlations of elements, mainly content modules. SCRs are located in a logic layer between content management systems, semantic modelling systems and content delivery portals (CDP). SCRs form ontologies that consist mainly of untyped semantic relationships between metadata classes or entities [6]. The correlation rules offer the possibility of providing so-called micro-docs in content delivery and other information portals. microDocs solve the problem of providing information, either by providing an overabundance of information or by displaying information without the necessary context. By connecting data via SCR, it becomes possible to pass on an information need directed at one source to other sources via semantic connections. In this way, it is possible to find documents that are not primarily assigned to the information need, from a subjective point of view. microDocs can be understood as a conceptual basis for an improved and contextually enriched provision of information. SCRs then represent one of the technical implementations of microDocs [6].

To create SCR, two classes are created, one for the InRules and one for the OutRules. InRules are the primary objects that are assumed. The OutRules are the secondary objects that can be addressed from the InRule. The classes are assigned certain properties that they pass on to their instances. A correlation property links the primary objects addressed by the InRule with the secondary objects via the respective OutRule. The SCR is a high-level ontology with the classes InRule, OutRule and the relations [6].

3 Concept and implementation

3.1 Concept objectives

The first and one of the most important processes for the creation of an ontology is requirements management. In this phase, the target and scope definitions are set. In order to be able to define the requirements at all, the exact use case was defined beforehand. The use case to be processed decides on the queries to request the information and the queries decide on the type of model to be modelled. The main goal of this work is to create an ontology for the company about their product world and to simplify information management and retrieval via further semantic technologies.

The definition of the scope of the ontology is defined by the objectives set: The creation of an ontology to represent the release of child seats and specific car seats and to replace the specific type list. Although the ontology should be able to represent further use cases in the later course, it is advisable when setting the objectives to first focus on a few concrete goals. Otherwise, difficulties may arise during the evaluation, and it may not be possible to check exactly whether the ontology completely fulfils its purpose.

3.2 Creating a database

Up to now, the data of the products, the dependencies among each other and the dependencies to the vehicles were maintained in various excel tables. For the different use cases in the company, the same information was often managed in several places, as similar information was needed for each use case, but not the same information. This resulted in an enormous maintenance effort within the individual departments. In addition, new information had to be changed or entered in different places. The necessary information was extracted from the tables and compared. An overarching table was created from the various Excel tables, into which all information was entered multiplied to the maximum. The Excel table represents a relational database with unique entries on the relationships of a specific vehicle seat and product and the relationships of a child with specific characteristics and the products.

3.3 Modelling the ontology

Within the project, the ontology modelling was implemented in the tool Protégé. Protégé is a software system used for creating and editing ontologies. It was developed by the Stanford University Center for biomedical Informatics Research. Protégé offers possibilities for mechanisms to define relationships between classes, entities, and properties. The open-source tool has an editor that is used for modelling the ontologies. Depending on the use case, various plug-ins can be included as extensions. Consequently, individual adaptations are possible.

3.3.1 Model approach I

The modelling is implemented in two different approaches to evaluate the use of the ontology in the best possible way. The first approach is strongly oriented towards the table-based database. A row of the table writes that there is a relationship between the individual entries in the cells both horizontally and vertically. The vertical entries (columns) each form a class with the entries themselves as entities due to the same characteristics. The rows of the table map the interdependencies of the individual entities and thus also form a class (rules). The entities of the Rules class form the rule world of the ontology and are named via unique IDs. The implementation leans on the idea of a relational database model, which allows for validity and affiliations based on keys.

The first approach is intended to show the most easily implementable modelling of an ontology. Therefore, the relationships are not named by intention, but are all represented by the same Object Property Selects. The relationship only describes that there is a connection between two objects. For the description of an application domain for a human interpretation, this kind of modelling would be insufficient, as hardly any information is provided. For purely machine processing, this approach can be pursued further.

Defined relations must be described in more detail via domain and range restrictions. For each relation, it must be defined what kind of objects can be potentially connected. Properties link instances of a domain to instances of a range to identify the arguments of a relationship. Range and domain are classes or trunks of classes that are specified as the first and second arguments of binary relations.

Fig. 3. Domain and ranges of the *Selects* object property

Data properties in OWL are used for linking entities with literals. Therefore, all information on years, such as start and end year of the construction years of the cars, were created as data properties. This approach resulted in unnecessary complexity in the queries, which did not correspond to the idea of modelling as simply as possible.

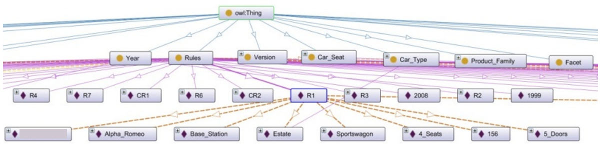

Fig. 4 Model approach I (simple)

3.3.2 Model approach II

The second modelling approach shows a significantly higher complexity and semantic richness. In the implementation, the possibility of ontologies was used that non-linear dependencies can be modelled. The investigation of different data sources has already been carried out for the first model. Therefore, the table created could also be used as a data basis for model II. In the first step, the information was put into an organised form by means of visualisation. This made it possible to directly identify the classes, their entities, and relationships.

This is followed by the formalisation phase. This involves the transformation of the visual representation into a formal coded ontology using OWL. This means that just like in Model I, the classes are created with the respective entities. One problem with the modelling was that, despite this, the use of keys could not be dispensed with. The dependency between car and product does not depend on a specific car, but on the individual car seats. However, since the seats do not have their own designations, they were given IDs and can be determined via the properties of the car.

In contrast to the first model, the relationships are not defined on a linear basis per rule, but the real relationships are created among each other. Classes are described by the relationships of the entities that belong to that class. Due to the dependencies of the objects among each other shown above, there are no binary dependencies, but always relationships of three objects. To solve this problem, the attachment methods were modelled as a connecting relationship (object property). This creates the problem that there can be two different relationships from a specific initial object to a second object. This type of modelling is a rather unusual procedure but is necessary for the description of the use case.

Essentially, the interpretation of groups is restricted by giving them more precise descriptions. Restrictions, which are basically logical axioms, compile meaningful descriptions and definitions for the classes and their instantiation as individual classes.

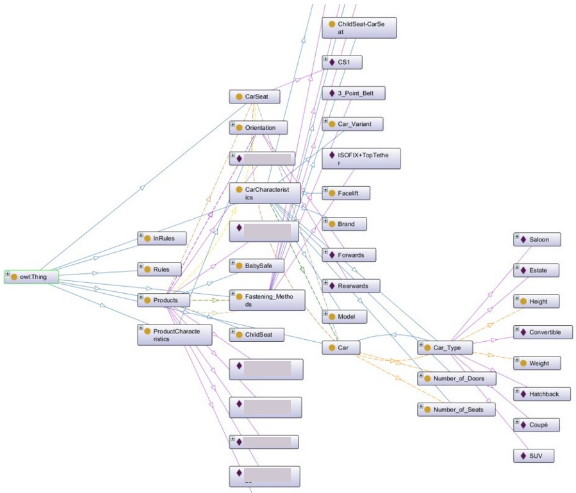

Fig. 5 Model approach II (complex)

3.4 Writing queries

Since the first modelling approach was developed in the first step and the more complex approach was only implemented in a later phase of the project, the queries for Model I were developed. Basically, a SPARQL query consists of six parts. At the beginning are the base and prefix declarations (I). In the declaration, base and prefix IRIs can be declared, which can be used in the triples to abbreviate IRIs. The name of the prefix is not important. A colon was chosen for the query created. (PREFIX : <http://www.semanticweb.org/sophie.bischoff/ontologies#>).

This is followed by the construction of data sets (II). For the development of the queries in Protégé this step was not necessary, because the queries always refer to the opened graph only. In the productive environment imaya (see chapter 4.1), the graphs are identified by a unique ID that must be specified in the construction.
The third part stands for the query type (III). SPARQL supports four query types: SELECT, ASK, CONSTRUCT and DESCRIBE. All of them return different types of results. SELECT was used for the queries in the research paper. In the case of SELECT, solution modifiers allow the control of solution duplications. These can be excluded in the result list with the modifier DISTINCT. Using the modifier REDUCED only allows duplicate solutions to be removed but does not guarantee this.

The WHERE clause (IV) specifies the patterns that are queried in the graph to generate solutions. The query refers to three variables that must be queried. First, the selects of the model by vehicle and product are queried (?selects), and second, the common selects from this model and the model to the child and product (?selectGem). In addition, a variable must be introduced that asks for the year (?year), since the years were modelled via data properties. The variables can be restricted as desired by specifying entities (e.g. :Audi : A1).

The aggregation (V) specifies variables or other expressions by which the solutions can be filtered depending on the result of the aggregation. The results are filtered by the constraining entities. Since all relationships were selected using the same Selects Property, these would otherwise also always be displayed in the result.
The solution modifiers (VI) indicate how the solutions should be ordered. GROUP BY orders the results according to the variable specified first, according to the selects.

```
## I___________________________
PREFIX :  <http://www.semanticweb.org/sophie.bischoff/ontologies#>

## II__________________________
FROM URI

## III_________________________
SELECT DISTINCT ?selects ?selectBoth ?year

## IV__________________________
WHERE {
        ?r :Selects     :Product,
                        :Audi ,
                        :A1 ,
                        ?selects ,
                        ?selectBoth ;
                        ?year .

?b :Selects             :61-105_cm ,
                        :Product ,
                        ?selectBoth .

## V___________________________
FILTER (?selects != :Product
        && ?selects != :Audi
        && ?selects != :A1
        && ?selects != ?selectBoth
        && ?selectsBoth != :Product)
}

## VI__________________________
GROUP BY ?selects ?selectBoth ?year
```

Fig. 6 SPARQL query for the Product Knowledge Service

4 Applications

4.1 Product Knowledge Service

The Product Knowledge Service is primarily intended to replace the classic user manuals and specific type lists. The service is a combination of CDP and interactive instructions. The type lists contain information on which child seat can be attached to which car seat using which attachment method. Previously, the user was instructed in the instructions for use to check the specific type list to see which car seats the child seat may be attached to. The user therefore had to retrieve the information from an external document. This significantly disturbs the natural reading flow and affects the user experience.
imaya is a software for information management from the company proricon. It provides solutions for centralized management, provision, search and exchange of information. In this research project, imaya's graph database was used. For this, the ontology in OWL format was exported from Protégé and uploaded to imaya. The modelled statements are stored in the graph database in the triple pattern and can thus be queried via the created queries. imaya has a SPARQL interface to execute the queries directly in the tool. For the Product Knowledge Service, an interface was created in which the queries are embedded in the backend so that the interface can be made available to users. The interface provides dropdown fields that are populated with the contents of the ontology sorted by class. The user can enter information about his car, his product, and the child to be mounted and gets the information which installation is allowed with which car seat and with which specific fastening method.

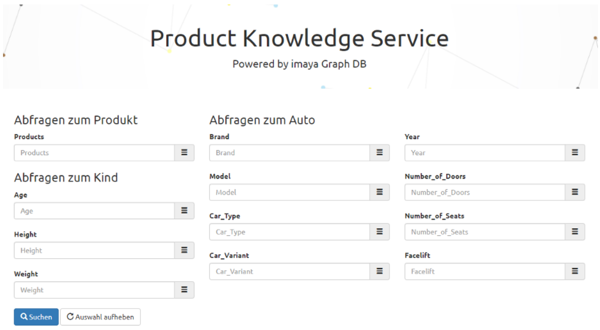

Fig. 7 Frontend of the Product Knowledge Service

4.2 Sales interface

Sales logic can also be mapped using the logic created in the Product Knowledge Service. The background behind selling a product is that a customer already has a child and often a car. All three components are already contained in the ontology and linked to each other with the appropriate relationships. So, the customer can select his specific car and provide information about the size, age and weight of the child and receives information about suitable child seats for this combination. In the further procedure, a separate user interface would have to be built for this use case, since the Product Knowledge Service also provides information that is not yet relevant for the purchase. In addition, product images might have to be stored in the sales logic. In summary, however, it can be said that the ontology is used as a common knowledge and data basis. Also, the generically built queries, can be adopted for the use case of the sales logic.

4.3 Content delivery

The knowledge service should not be limited exclusively to technical documentation content. The system should serve as a central information portal. In addition to the operating instructions and information from the type lists, the user should also be offered suitable service or marketing content. The focus will be on the operating instructions. However, the aim is also to offer users all the information on a topic that they need for maximum understanding of the subject. The content will be related to other information. The linking of content is achieved through a classification concept using SCR.

Use cases from the consumer service environment are provided. Based on semantic and linked information, these can then be analytically modelled and optimised. To define user-centric use cases, various aspects must be considered. These are relevant and defined for the use cases in the context of this research and include the definition of the user's role, their situation and their prior knowledge or experience.

SCRs are modelled using a lightweight ontology and can be integrated into the existing ontology. For this, the required metadata from the existing metadata classification is used for the semantic classification in the existing CCMS. The metadata is implemented as a hierarchical structure with classes and entities. In addition, the classes for InRules and OutRules are created and the relation *hasCorrelation* to be able to specify the relationships among each other. The entities of the class form the individual InRules and OutRules. The relevant metadata is assigned to these via select relations, via which content can be requested from the CCMS. The queries are formulated via SPARQL. The InRules are found by specifying metadata and the required OutRules can be output from them via variables. The metadata, according to which the matching secondary objects filter, is attached to the OutRules.

4.4 Administration interface

A knowledge representation must be flexible and be able to adapt to changes. Ontology developers are already going in the right direction with the topic of ontology evolution, however, besides some methods, there is no widespread implementation of a tool to support this evolution yet. The situation is the same in the related area of quality assurance of ontologies. When a knowledge representation is open to constant change, it is even more important to be able to ensure high quality at any time. But this area is also far away from standards. The overall goal of the automation of processes should also be achieved at the level of administrative tasks. The idea for the implementation is to create a web interface on which the creation, modification and maintenance of data records can be executed. This extended maintenance interface represents another use case for the created ontology in interaction with SPARQL.
Standard maintenance transaction and other functions will be bundled and made accessible through one interface. The interface will be exclusively accessible for the internal use of the Technical Communication (TC) department. The TC will thus benefit from ease of use and easy comprehension of data maintenance. The goal of a web maintenance interface is that the system can be operated without having to undergo intensive training. Like the Knowledge Service, the interface will be structured as an HTML page. The HTML forms trigger Java Script functions, which in turn trigger SPARQL queries. The SPARQL operators for updating a graph can be used to create, update, or delete elements in the form of triples.

5 Findings and outlook

5.1 Chances and limitations of ontologies

This article examines to what extent the introduction of semantic technologies in a company can be considered useful, especially in technical communication. Given the different use cases tested in the research, it can be summarised that ontologies are a good tool to represent knowledge and use it as a central source. In addition, it is possible to check via the ontology whether data records already exist for certain vehicles. This allows duplicates to be removed from the systems. The problem with using an ontology is that the necessary competences must first be created within a company.

When modelling an ontology, there is basically no right or wrong. However, once a type has been decided upon, it must be urgently maintained consistently in order to be able to exclude sources of error.
When introducing an ontology into a company, it is initially time-consuming and also costly, which can have a deterrent effect. However, an ontology ultimately reduces effort, allows errors to be avoided and offers great potential for automation.

5.2 Complexity of queries

The complexity of the queries is not necessarily related to the complexity of the ontology. In this use case, an ontology was queried that was completely modelled using the same relationship. However, this resulted in the problem that it was not possible to query specifically for a class without always outputting all other endpoints of the *Selects* relationship. Whether the queries can be designed more explicitly with Model II is to be examined

5.3 Qualifications for technical writers

Due to the advancing development within technical communication, the professional field is changing in terms of the required competences. Even the name 'technical writer' is no longer necessarily a good description. The term 'knowledge architect' has become more appropriate. Essential skills are the technical know-how on practical methods of knowledge modelling, which encompasses a whole methodology. Business analysis is a good skill to have as a knowledge architect. A knowledge architect should know how to interpret, represent, and communicate concepts, states, processes, workflows. A skill that has been required of classical technical writers is good writing skills and mastery of the language in which the ontology concepts are to be communicated. This is necessary to produce high quality ontology documentation. A knowledge architect develops a very keen eye for detail. The work of grasping and representing concepts requires meticulousness and multitasking ability.
There is a lot of overlap between these skills and those of the "business analyst" and the "enterprise architect" whose job it is to capture domain models from different perspectives in a collaborative environment.

5.4 Outlook

The next steps of this research work would be to further elaborate the individual approaches in the first step and to carry out prototype implementations. In particular, the creation of an administration interface for the creation of a cleansed database is a high-priority process.

Further scenarios for the application of the ontology would be the use as a terminology management tool. The metadata and ontology have so far been created exclusively in English. However, many technical terms are used in it and their names should also be available in other languages. In addition, ontologies can be used to determine preferred and prohibited terms to improve consistency within texts.

Furthermore, it would make sense to store an inheritance logic by which several data sets are created for release. So far, however, there is no process in the company that specifically shows the factors on which the releases depend. If one could create a guideline that determines whether a car seat and a child seat is released based on certain constructional factors, it would be possible to pursue the complex modelled approach of the ontology and to model the dependencies. However, since up to now each seat has been tested individually, it makes sense to continue using the model approach corresponding to a relational database (Model I) as a basis.

Acknowledgement

The author would like to thank Julian Stähle for the great collaboration, insightful discussions and for his support. I would also like to thank Prof. Dr. Ziegler for the opportunity to take part at ETLTC.

References

1. Sikos, L. *Mastering structured data on the semantic web. From HTML5 microdata to linked open data*. Apress. New York. (2015)
2. World Wide Web Consortium. *RDF.* Online : https://www.w3.org/RDF/ , 14.03.2022 10:00
3. Domingue, J. Fensel, D. Hendler, J. A. *Handbook of Semantic Web Technologies.* Springer. Berlin, Heidelberg. (2011)
4. Dengel, A. *Semantische Technologien. Grundlagen. Konzepte. Anwendungen.* Spektrum Akademischer Verlag. Heidelberg. (2011)
5. Jakus, G.; Milutinovic, V; Omerovic; S. Tomazic S. *Concepts, Ontologies and Knowledge Representation.* Springer (2013)
6. Ziegler, W. *Semantic Correlation Rules as a Logic Layer between Content Management and Content Delivery,* Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2021)
7. Ziegler W. *Extending intelligent content delivery in technical communication by semantics: microdocuments and content services*, Proceedings of the ETLTC ACM Chapter International Conference. Aizuwakamatsu, Japan. (2020)