Lexicalized ontology for a business rules management platform: An automotive use case

Nouha Omrane, Adeline Nazarenko, Peter Rosina, Sylvie Szulman, Christoph Westphal

HAL Id: hal-00625006
https://hal.archives-ouvertes.fr/hal-00625006

Submitted on 20 Sep 2011

Lexicalized ontology for a business rules management platform:
An automotive use case

Nouha Omrane*, Adeline Nazarenko*, Peter Rosina**, Sylvie Szulman*,
Christoph Westphal**

* LIPN, Université Paris 13 & CNRS (UMR 7030), France
** AUDI AG, Germany

Abstract. This paper describes a platform that helps industrial domain experts to preserve the connection between textual sources and formalized business rules by using lexicalized ontologies both for links and for storage of the conceptual knowledge.
Business Rules Management Systems (BRMSs) are used to update and query business rules of an automotive use case. They rely strongly on domain ontologies, which model the business knowledge and provide a conceptual vocabulary for the formalization of the rules that are expressed in written policies. We show that lexicalized ontologies are a key component of such BRMSs and how such knowledge can be encoded.
Our proposed solution supports domain experts in the automotive industry in understanding and maintaining their business rules by presenting the relevant source documents that were used to create the ontological concepts. The use case is based on a car development scenario that models the connection between car testing scenarios, e.g., safety tests, and the methods and tools used to analyze and prepare these tests. The intended solution has been developed in the ONTORULE project and is still work in progress.

Keywords: Business rules, domain ontology, semantic annotation

1 Introduction

Business Rules Management Systems (BRMSs) are software applications that help organizations to separate their application code from their business knowledge. BRMSs help the users to author and maintain business rules and apply decision logic that reflects this business knowledge. The business rules can have different origins, such as regulations, policy documents or business logic directly entered by domain experts. This business logic expresses both development processes and coherence between different events, including conditions and the resulting conclusions.

One of the main advantages of expressing the logic in business rules is that the domain knowledge is independent of the application code that uses this logic. In such way, there is no need to alter the application code itself, when business logic evolves, new policies are applied, already introduced policies change or retire. Thus, the use of BRMSs leads to increased flexibility and agility of the organization.

In the ONTORULE project, Audi uses OntoBroker as the execution environment for its business rules which are formalized in Objectlogic[1]. The OWL representation of the domain ontology has been developed in parallel for the research purposes described in this paper.

Domain experts who are not also business rules experts may have difficulties expressing their knowledge in formalized logic languages. Supporting them in their management of the knowledge needed to write these rules is one of the goals of the ONTORULE project.

We propose building an ontology as a formal model for representing conceptual vocabulary that is used to express business rules in written policies. Using a normalized vocabulary helps domain experts in writing rules more efficiently and is less costly than managing controlled vocabulary. Such ontologies are shared conceptual models, so experts can share the same vocabulary. We use the OWL-DL language to represent concepts and properties of the domain ontology. In addition,, the ontology is linked to the lexicon used to express rules in the text, so experts can query source documents. This calls for a formalism to link linguistic elements to conceptual ones. We opt to use the SKOS[2] language which provides basic elements to link domain concepts to terms from the text. The combination of OWL entities, SKOS concepts and their related information form a lexicalized ontology which supports the semantic annotation of documents.

The paper is organized as follows. Section 2 describes the Audi use case and Audi's expectations of using a lexicalized ontology. Section 3 explains the choice of OWL and SKOS as languages to support the lexicalized ontology. Section 4 reports the experimentations. Section 5 describes the related work in linking ontology to lexicon and semantic information retrieval. Section 6 presents the conclusion and future work.

2   The Audi BRMS or platform

Nowadays, the development of new cars has become very challenging and many different process steps are involved. Computer Aided technologies (CAx), like virtual modeling, simulations or the analysis and planning of physical testing, need to be integrated even tighter to satisfy the higher requirements and reduced time-to-market which also shortens the development cycles.

---

[1] Ontology, rule and query language introduced by ontoprise as successor of F-Logic. Main language supported by the semantic web ontology repository and inference engine OntoBroker.

[2] Simple Knowledge Organization System

In the ONTORULE project Audi is developing a prototype application that makes use of ontologies and business rules that includes and visualizes the context of the following use case.

The development of a car typically follows a process starting with product planning, runs over concept development to testing (virtual & physical) until a car can be launched for production. This long process is strongly supported by different Computer Aided technologies, e.g., Computer Aided Design (CAD), Computer Aided Engineering (CAE), and Computer Aided Testing (CAT):

- CAD: This branch provides methodologies for the virtual design of the parts of a car - e.g., digital mock-up (DMU) methods, parametric design methods, - and verification of the design concerning the geometry when integrated into a car.
- CAE: The main task of CAE is to provide methodologies for simulating the behaviour of a car and its functions - e.g., finite element analysis (FEA) for crash simulation, computational fluid dynamics for thermal management, and multi body simulation (MBS) for driving dynamics.
- CAT: Methodologies for performing physical tests of cars are provided by CAT - e.g., vehicle management, job control, testing control, and test result analysis.

Ontologies together with business rules help Audi to keep abreast of technology advances and use them in its R&D IT applications. Especially the interweaving of the various CAx technologies will help Audi to reduce development time and cost.

One of the first steps in the development cycle of a new car model is the definition of properties, i.e., features that can be experienced by customers, like driving comfort, safety or sportiness that the desired car has to fulfill. These target properties are listed in catalogs, that describe the car's required and mandatory behavior in various granularities.

At a later stage in the development cycle, engineers begin to design new or modify already existing Computer Aided Design models, that have to comply with Audi's high quality expectations. These models are then used in simulations (Computer Aided Engineering) or serve as a model for the physical parts which are tested in Computer Aided Testing. For example, new electronic components, like an Electronic Stability Control (ESC), are tested in Hardware in the Loop (HiL) simulations that make use of virtual models that behave like the related dynamic systems.

The entirety of all these attempts and approaches, physical and virtual, is called CAx Methods. The different CAx Methods ensure either that the desired properties are achieved, or that legal requirements are fulfilled. Components, such as ESC, are referred to as "Solution Concepts" (cf. Fig. 1). They affect the customer experience or vehicle property directly.

The knowledge about the initial relation between a property that was defined in the beginning of the development cycle and a CAD part with its related CAx Methods often perishes during the progression of the vehicle's design and

Fig. 1. Simplified extract of the intended Audi R&D ontology
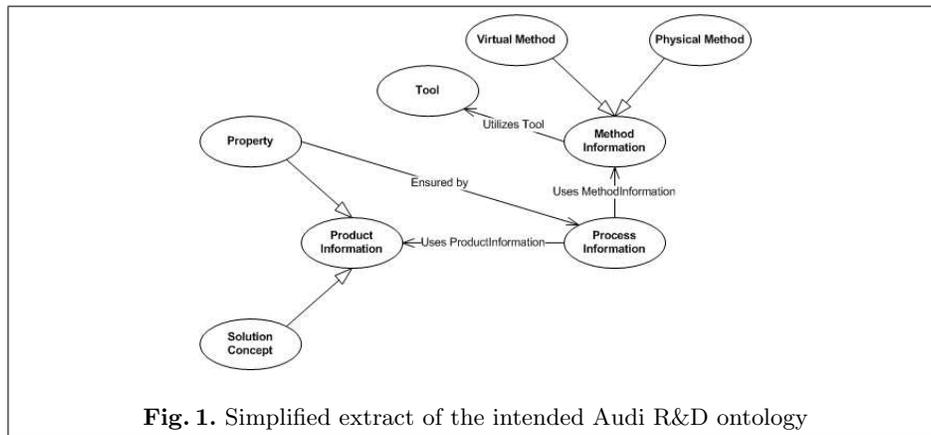
construction. In addition, requirements and targeted properties may change over time.

By modeling the described scenario in an ontology, and formulating and processing the relevant rules in a BRMS, Audi expects to reduce the knowledge gap between the various process steps for the involved employees and departments and thus to benefit from a tighter CAx integration. Sharing the knowledge by using a common tool and shared data will reduce time-consuming data acquisition and ensures that the personnel involved will access identical data which will help speed up the development and innovation cycles [7].

For example, with the help of business rules, we are calculating the duration of processes (*processDuration*) that involve different CAx methods (cf. rule example below). Every CAx Method has an assigned attribute for either an estimated or an actual value for its lead time, cost and maturity. Testing if the requirements for one vehicle property are fulfilled normally takes several process steps. As an example, the execution of a CAE simulation may take only minutes - the preparation, the modeling, and so on might take several weeks.

Every process, e.g., an analysis or the validation of such an analysis, makes use of a CAx Method (cf. Fig. 1). The function *processDuration* calculates the whole process duration: the relevant attribute *time* is queried with the function *methodDuration* (which is defined in another rule) from all the methods involved in the relevant subprocesses.

With this rule, the application is able to visualize different durations involving different methods that all test one specific vehicle property. These indications will support managers when planning their projects, so that they can choose whichever process is best suited for their work.

```
processDuration(?X,?Y,?totalTime) :-
    methodDuration(?X,?Y,?totalTime).
processDuration(?X,?Z,?totalTime) :-
```

```
methodDuration(?Y,?Z,?time) AND
processDuration(?X,?Y,?previousTime) AND
?totalTime = ?time + ?previousTime.
```

(Rule example, that calculates the total process duration.)

One of the difficulties with business knowledge rules is that various departments or roles sometimes use different vocabularies for the same things so they cannot understand each other immediately. Addionally, formalized rules per se are often not easy to understand.

Using an ontology as a unified model for a heterogeneous vocabulary and annotating the rules and the underlying ontology has another advantage: it will reduce misunderstandings and ensure that people are discussing the same thing. Also, the users can easily confirm and verify the appropriateness of the modeled semantic relations. The prototype that is to be developed will handle links between source documents, such as policies and internal documents, i.e. policies or internal documents, and the concepts and instances of the ontology. Also, if a business rule originates directly from a legal document, the relevant passage will be linked to the rule in the same way.

3 The role of the lexicalized ontology in BRMS

A BRMS of the kind described here relies strongly on domain knowledge that is encoded in a lexicalized ontology. This section shows how OWL-DL and SKOS standards support the needs for formalization, document annotation, normalization and documentation that business experts face when designing a business rule application.

3.1 Formalization of domain knowledge

The idea of the ontology as a conceptual model was explicitely introduced by [6] affirming that an ontology is: "the specification of a conceptualization". The ontology is a set of concepts that are formally defined. We opted for the for OWL-DL[3] language to express domain concepts or classes and their properties.

Such an ontology not only gives the vocabulary to be used in expressing the rules, it also provides a structured vocabulary that encodes relationships between concepts and supports cheking for inconsistencies. The following example describes the concept *BuckleTest*, its related SKOS concept

```
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#BuckleTest">
```

having as label "buckle test" in the text:

---

[3] http://www.w3.org/TR/owl-guide/

```
<owl:Class rdf:about="&onto;#BuckleTest">
  <rdfs:subClassOf rdf:resource="&onto;#PhysicalMethod"/>
</owl:Class>
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#BuckleTest">
  <skos:prefLabel>buckle test</skos:prefLabel>
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/
    core#Concept"/>
</rdf:Description>
```

Formalizing the business rule vocabulary in an ontology gives a structure to, and enables querying of, the rule base. For example, we can display all the roles involving physical methods by querying only the parent concept of all physical method concepts.

Experts can also query the ontology itself to search, e.g., for a test that verifies a property and is related to some constraints, as soon as these properties are encoded as concept roles. The following example shows a query written in ObjectLogic to query the ontology. The concept *ProductInformation* is used in the verification process (*ProcessInformation*) and every *ProcessInformation* uses a designated method (i.e., physical or virtual method). The concept *MethodInformation* has as subconcepts all the physical and virtual methods (cf. Fig. 1). The result of this query displays all *ProductInformation* (i.e., car parts, functions, etc) and the tools that use them.

```
@{Systemanalysis_ManagedProductInformation, options[outorder(
  ?Tool,?ProductInformation),fillNull]}
?- ?MethodInformation:MethodInformation[utilizes_Tool->?Tool]
  AND ?ProcessInformation:ProcessInformation[
  uses_ProductInformation->?ProductInformation,uses_Method->
  ?MethodInformation]
  AND ?ProductInformation:ProductInformation.
```

Finally, using the OWL-DL language supports reasoning on the ontology. This is useful for searching for hidden information that is implicit in the rules, for inferring new knowledge, updating the rule base and ultimately improving the business of the organization. For example, experts may recognize that a safety test is less costly with some specific parameters.

For example, the concept *SeatBelt* which describes the seat belt, is related through the role *assuredBy* to the tests *CorrosionTest* and *SeatBeltFlipTest*, which are used to test its safety:

```
<owl:Class rdf:about="&onto;#SeatBelt">
  <assuredBy rdf:resource="&onto;#CorrosionTest"/>
  <assuredBy rdf:resource="&onto;#SeatBeltFlipTest"/>
</owl:Class>
```

3.2 Semantic annotation of documents

A key issue for experts in managing a rule base is to recognize that the meaning of formal rules and natural language sources, such as written policies and documentation is a precious source of information. It is also important to update the business rules as organizations often modify their policies according to internal or external constraints.

It is therefore important to be able to mine textual sources to understand how a given concept is used in business documents, what rules are related to it and how those concepts and rules evolve when the policies are updated. This is achieved through the semantic annotation of the documents in which the mentions of the ontological entities (concepts, instances and roles) are highlighted and can be searched for.

Semantic annotation means that ontological entities are related to the terms that can be used to mention them in the texts and calls for designing lexicalized ontologies. When the ontology has been created from textual source, as for the Audi ontology, it is easy to keep track of the terms that denote the various conceptual entities. The resulting lexicalized ontology is used to annotate source documents and to query them.

Our aim is to save the terms related to the conceptual vocabulary that is used to express the business rules. We don't need to encode sophisticated information such as the morphological structure of terms since we do not perform a deep analysis of the documents. We simply need to save the various linguistic units that denote a concept, instance or role. We use SKOS for that.

SKOS supports encoding of *SKOS concepts* that represent the links between the OWL concepts and their related terms, which are encoded as *skos labels*[4]. This relation is described by <rdf:Description rdf:about>.

3.3 Normalization of vocabularies

When designing and updating business rules, experts face the problem of the heterogeneity of information sources and multilingualism. *SKOS* also supports that normalization of vocabularies.

SKOS enables association of a given SKOS concept with the various terms or labels that denote it in the texts or any other information source. For a given concept, *SKOS* supports distinguishing one preferred label and as many alternative labels as necessary, using the <skos:prefLabel> and <skos:altLabel> properties. In the Audi ontology, for example, the SKOS concept *LowTemperatureChamber* is linked to two terms: *low temperature chamber* is encoded as the preferred label and *refrigerated cabinet* as its alternative form:

```
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#LowTemperatureChamber">
  <skos:prefLabel>low temperature chamber</skos:prefLabel>
  <skos:altLabel>low-temperature chamber</skos:altLabel>
```

---

[4] http://www.w3.org/2004/02/skos/

```
    <skos:altLabel>refrigerated cabinet</skos:altLabel>
    <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#
      Concept"/>
</rdf:Description>
```

Alternative labels are used to encode linguistic variants (e.g., seat belt which is a belt) or unify different vocabularies (e.g., low temperature chamber actually has the same meaning as refrigerated cabinet). The following example shows that the SKOS concept *SeatBelt* is related to two terms *seat belt* and *belt* in the text:

```
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#SeatBelt">
  <skos:prefLabel>seat belt</skos:prefLabel>
  <skos:altLabel>belt</skos:altLabel>
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#
    Concept"/>
</rdf:Description>
```

SKOS also supports the encoding of multilingual information. The information about the language used is described by <`rdf:lang=''en''`>. For example, the SKOS concept *TrolleyTest* has a preferred label "trolley test" which is mentioned in English in the text and an alternative label "Schlittentest" in German.

```
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#TrolleyTest">
  <skos:prefLabel df:lang="en">trolley test</skos:prefLabel>
   <skos:altLabel df:lang="de">Schlittentest</skos:altLabel>
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#
    Concept"/>
</rdf:Description>
```

Thanks to the alternative labels and language tags, *SKOS* therefore helps experts managing the heterogeneity of the vocabulary of their sources and controlling the vocabulary used for designing rules.

3.4 Documentation of the shared knowledge

Since experts often have to manage a large volume of information but do not always formally describe all the concepts, it is important to add informal documentation when it is available. Defining concepts in natural language is very important to understand what concepts mean, especially if they have ambiguous or implicit labels.

Since legal documents such as policies often define their terminology precisiely, we propose to extract those definitions from the source documents when designing the ontology and to associate them with the related SKOS concepts using the label <skos:definition>.

Source documents are exploited to find definition for existing concepts in the Audi ontology. For example, the concept *ReferenceZone* is described as follows:

```
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#ReferenceZone">
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#
    Concept"/>
  <skos:definition>"Reference zone" means the space between two
    vertical longitudinal planes , 400 mm apart and symmetrical
    with respect planes , 400 mm apart and symmetrical with
    respect to the H point , and defined by rotation from
    vertical to horizontal of the head form apparatus.
  </skos:definition>
  <skos:prefLabel>reference zone</skos:prefLabel>
</rdf:Description>
```

3.5   Formalism for the Audi lexicalized ontology

The Audi ontology is a formal representation of the conceptual vocabulary used to express business rules in written policies. In the Audi use case, we use OWL-DL to describe concepts and their roles. Structuring the vocabulary and normalizing it supports querying of the ontology in order to manage the knowledge base, infer new knowledge and detect inconsistency. As the Audi ontology is lexicalized, the domain concepts and their occurrences in the text can be matched onto one another thanks to the linkage of OWL entities, SKOS concepts and labels.

This is a simple efficient way to represent lexicalized ontologies and we show in the following section its benefit for the Audi BRMS. Figure 2 describes how the Audi ontology is linked to the lexicon and annotated text.
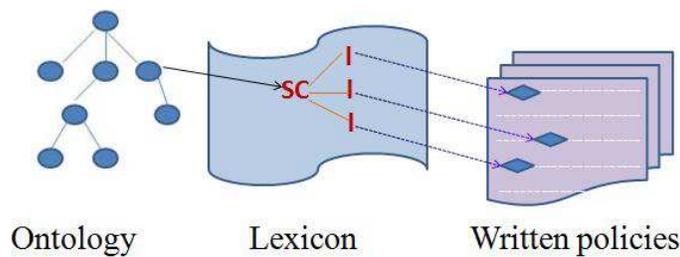

Fig. 2. A lexicalized ontology for annotating source documents. Each concept from the ontology is linked to a SKOS concept *SC* and each SKOS concept is related to its labels *l*. The annotations link some text entities to these labels

4 Experiments in the Audi use case

This section presents the Audi ontology and illustrates the benefit in the Audi use case of having such a lexicalized ontology.

4.1 The Audi BRMS ontology

The Audi ontology has been built in two steps. At first, the goal was to integrate the various existing knowledge sources in a single one. This resulted into a small conceptual model (around 30 concepts) associated with a large knowledge base (thousands of instances).

In a second step, in order to better fits the experts' needs for semantic querying and document mining, the initial ontology has been restructured and lexicalized. It also appeared useful to increase the granularity of the domain model so as to represent not only the various types of tests but also their actual occurrences in the car manufacturing process (instances that are related to the different tests applied to specific vehicle models).

This led to encoding of various elements as concepts rather than instances (90 concepts were added). Modeling tests as concepts supports, for instance, querying of the Audi ontology in such a way as to detect implicit relationships between tests, tools and parameters, which are important for the safety of vehicles. The conceptual structure has been reorganized (4 subsumption levels instead of 1). A SKOS resource has been associated with this resulting ontology: each concept is related to at least 1 preferred label and up to 5 alternative labels. In addition, using a subset of the initial ontology for the exploration of written policies showed that 10 of the mentioned tests were missing in the initial ontology and led us to enrich it [9].

4.2 Semantic querying

The knowledge base of our ontology currently consists of several thousands of instances separated into more than 30 concepts. It currently reflects only a small part of the whole use case and will be enhanced over time. We have annotated a subset of these instances in English and German to allow the users to use the aimed-at application in their preferred language. Additionally, alternative labels can be added when different users prefer different terms in their daily work.
We have also planned an interface to query the ontology and search for specific instances and concepts. For that reason, an annotated ontology is much more appropriate.

4.3 Document mining

We enrich the Audi ontology with new concepts and link conceptual elements to linguistic ones from the Audi documents. We obtain a lexicalized ontology that contains 90 SKOS concept, 90 preferred terms and 16 alternatives labels.

For example, we create the SKOS concept *BreakingStrengthOfStrapTest* that describes a specific test of the seat belt and link the preferred and alternative labels which are mentioned in source documents.

```
<rdf:Description rdf:about="http://lipn.univ-paris13.fr/RCLN/
    terminae/Audi#BreakingStrengthOfStrapTest">
  <skos:prefLabel>breaking strength test</skos:prefLabel>
  <skos:altLabel>test of breaking strength</skos:altLabel>
  <skos:altLabel>breaking strength of strap test
  </skos:altLabel>
  <skos:altLabel>test of breaking strength of strap
  </skos:altLabel>
  <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#
    Concept"/>
</rdf:Description>
```

Once the ontology is lexicalized, domain experts can query source documents to search for fragments of texts that describe specific concepts mentioned in rules. For example, they can find all references of the concept *BreakingStrengthOfStrapTest* in the text, wherever it is mentioned in the documents:

```
prefix schema:<http://lipn.univ-paris13.fr/RCLN/terminae/schema#>
prefix onto:<http://lipn.univ-paris13.fr/RCLN/terminae/Audi#>
prefix skos:<http://www.w3.org/2004/02/skos/core#Concept>
select ?sentence
where
{  ?sentence rdf:type schema:sentence
   ?sentence schema:annotatedBy ?concept
   ?concept schema:realized concept <onto:
     BreakingStrengthOfStrapTest>
   ?skos skos:skosConcept ?concept
   ?skos skos:preflabel "test of breaking strength"
}
```

We can also search for all sentences where the physical methods are mentioned in the text. As the concepts expressing tests are sub-concepts of the concept "MethodInformation", we query the text by searching about all subconcepts of "MethodInformation". The following example shows the query:

```
prefix schema:<http://lipn.univ-paris13.fr/RCLN/terminae/schema#>
prefix onto:<http://lipn.univ-paris13.fr/RCLN/terminae/Audi#>
prefix skos:<http://www.w3.org/2004/02/skos/core#Concept>
select ?sentence
where
{  ?sentence rdf:type schema:sentence
   ?sentence schema:annotatedBy ?concept
   ?concept schema:realized concept <onto:MethodInformation>
```

```
?conceptfils rdfs:subClassOf ?concept
?skos skos:skosConcept ?conceptfils


}
```

Thanks to the labels of concepts, the ontology can be used to annotate the documents. Figure 3 shows an example of texts where all the mentions of known concepts are emphasized.
This supports experts in browsing of documents.

Two belts or restraint systems are required for the buckle inspection, the low-temperature buckle test, the low-temperature test described in paragraph 7.5.4. below where necessary, the buckle durability test, the belt corrosion test, the retractor operating tests, the dynamic test and the buckle-opening test after the dynamic test. One of these two samples shall be used for the inspection of the belt or restraint system.

Fig. 3. A fragment of text annotated by the lexicalized ontology.

5 Related work

Many research activities have tackled the problem of linking an ontology to a lexicon. Two major areas are of interest.

The first is the NLP domain which aims at adding some semantic structure to a lexicon by linking its elements to ontology's elements. There are several ways to combine a lexicon with an ontology. We describe the most popular ones.

LMF[5] standard [5] aims at detecting the relation between the words used to express objects and their formalization in the ontology. The mapping is assured by axioms declared in the ontology. LMF has the merit of assessing large scale lexical resources. TMF[6] is a standard ISO that allows describing the terms of a lexicon within a formal language. The supported language is based on the definition of a meta-model and a set of features used to represent elements of the lexicon (e.g., related class, language, lexical information). $OLIF$[7] is an open standard for exchange format that allows to represent morphological, syntactic and semantic data categories. LMM[8] is a formal language described in OWL-DL. It integrates linguistic knowledge resources (e.g., FrameNet) and foundational ontology (e.g., DOLCE). It has the advantage of dealing with multi-lingual resources.

The other family tries to link an ontology to a lexicon by introducing models for representing linguistic information for ontologies. [11] defines a meta model to

---

[5] Lexical Markup Framework
[6] Terminological Markup Framework
[7] Open Lexicon Interchange Format
[8] Linguistic Meta Model

distinguish terms and concepts. Terms are linked to concepts through a semantic relation "denote". *LexOnto* proposed by [2] supports definition of the lexicon by using all the expresseiveness of OWL. *LexOnto* considers a term as an "OWL: class" and presents a meta model that supports linking of terms to concepts. The relation between the elements can be simple or complex (e.g., sub-category-of).

LingInfo [1] define a meta class to link the linguistic properties to the concept or to its Data/Object properties. The meta class contains the associated term, its language and the morpho-syntactic structure of the term. [8] introduces a set of annotation rules to link an existing ontology to its lexicon that is described in the text. The authors consider a standard OWL-DL ontology that they extend and, for each domain ontology, propose to define a set of annotation rules that link each concept to its linguistic representations in the text.

LIR [10] supports storage of linguistic information in a lexical ontology and linkage to domain concepts via ontology relations.

From a practical point of view, the choice of one model or another depends on the aimed application and the task. Our aim is to build a lexicalized ontology to allow annotating the technical documents and thus to help the expert in exploring documents by querying its set of annotations explained above. We use for that a W3C standard *SKOS* that links linguistic to semantic knowledge. *SKOS* introduces especially the properties: *skos:prefLabel* and *skos:altLabel* to link preferred and alternative terms to each concept in the ontology.

Usually users query documents by entering keywords that occur in the target documents in Information Retrieval (IR). But searching relevant information through a keyword-based approach very often provides limited results. Therefore, IR adopts a conceptual search-approach to capture the user needs on a semantic level [4]. Information retrieval uses ontologies as formal models to query knowledge data base [3]. It takes into account these keywords and tries to match them with their corresponding concepts in the ontology. To do so, ontologies need to be linked to a rich lexicon as in lexicalized formalism described above so that the result obtained matches the query of users.

The ability to link an ontology to a lexicon is very important for annotation of documents. Annotating documents supports linking of textual units to their corresponding concepts in an ontology, and showing the relationship between these concepts and the terms that are related to them. Such a lexicalized ontology helps the exploration of texts through semantic queries once they are annotated with the ontology.

6   Conclusion

The proposed integration of CAx systems will increase the flexibility of the development process, allowing Audi to meet the increasing market demand for product diversification.

This integration relies on the design of an application that is currently under development and is based on a BRMS.

Our approach for the acquisition and management of the knowledge embodied

in such BRMS relies on a lexicalized ontology which unifies and normalizes the various vocabularies and links the conceptual knowledge to the source policies and regulation written in natural language.

Using a lexicalized ontology enables experts to determine the most suitable CAx Methods from given functional requirements and to query sources documents.

These new approaches, standards and technologies are already partially integrated in some processes. During the next years Audi will continue to incorporate the ONTORULE platform in their landscape which will lead to even less time-consuming, cheaper and higher quality processes in the innovation and development cycles.

References

1. Buitelaar, P., Sintek, M., Kiesel, M.: A Multilingual/Multimedia Lexicon Model for Ontologies. In: ESWC. pp. 502–513 (2006)
2. Cimiano, P., Haase, P., Herold, M., Mantel, M., Buitelaar, P.: LexOnto: A Model for Ontology Lexicons for Ontology-based NLP. In: Proceedings of OntoLex - From Text to Knowledge: The Lexicon/Ontology Interface (workshop at the International Semantic Web Conference) (2007)
3. Davies, J., Duke, A., Kiryakov, A.: Information Retrieval, searching in the 21st century, chap. Semantic Search. Wiley
4. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced Information Retrieval: An ontology-based approach (2010), http://dx.doi.org/10.1016/j.websem.2010.11.003
5. Francopoulo, G., Be, N., George, M., Calzolari, N., Monachini, M., Pet, M., Soria, C.: Lexical markup framework: ISO standard for semantic information in NLP lexicons. In: Workshop of the GLDV Working Group on Lexicography at the Biennial Spring Conference of the GLDV (2007)
6. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. In: Formal Ontology in Conceptual Analysis and Knowledge Representation. Available as Stanford Knowledge Systems Laboratory Report, Kluwer Academic (March 1993)
7. Kiss, E., Albert, P., Korf, R., Rosina, P., J.Hoppenbrouwers, S.Njissen: Market Intelligence Report, D8.4 ONTORULE Project Deliverable. Tech. rep. (2011)
8. Ma, Y., Audibert, L., Nazarenko, A.: Ontologies étendues pour l'annotation sémantique. In: 20mes Journées Francophones d'Ingénierie des Connaissances (IC09) (2009)
9. Omrane, N., Nazarenko, A., Szulman, S.: Les entités nommées : éléments pour la conceptualisation. In: 22mes Journées Francophones d'Ingénierie des Connaissances (IC11) (2011)
10. Peters, W., Espinoza, M., Montiel-Ponsoda, E., Sini, M.: Multilingual and localization support for ontologies. Technical report, D2.4.3 Neon Project Deliverable. Tech. rep. (2009)
11. Reymonet, A., Thomas, J., Aussenac-Gilles, N.: Modélisation de ressources termino-ontologiques en OWL. In: Trichet, F. (ed.) Journées Francophones d'Ingénierie des Connaissances (IC), Grenoble. pp. 169–180 (July 2007)