Christophe Debruyne and Cristian Paulus Vasquez. Exploiting Natural Language Definitions and (Legacy) Data for Facilitating Agreement Processes. In D. Winkler, S. Biffl, and J. Bergsmann, editors, SWQD, volume 133 of LNBIP, pages 244-258. Springer, 2013

Exploiting Natural Language Definitions and (Legacy) Data for Facilitating Agreement Processes

Christophe Debruyne, Cristian Vasquez

Semantics Technology and Applications Research Lab (STARLab),
Vrije Universiteit Brussel,
Pleinlaan 2, B-1050 Brussels, Belgium
{chrdebru,cvasquez}@vub.ac.be

Abstract. In IT, ontologies to enable semantic interoperability is only of the branches in which agreement between a heterogeneous group of stakeholders are of vital importance. As agreements are the result of interactions, appropriate methods should take into account the natural language used by the community. In this paper, we extend a method for reaching a consensus on a conceptualization within a community of stakeholders, exploiting the natural language communication between the stakeholders. We describe how agreements on informal and formal descriptions are complementary and interplay. To this end, we introduce, describe and motivate the nature of some of the agreements and the two distinct levels of commitment. We furthermore show how these commitments can be exploited to steer the agreement processes. Concepts introduced in this paper have been implemented in a tool for collaborative ontology engineering, called GOSPL, which can be also adopted for other purposes, e.g., the construction a lexicon for larger software projects.

Keywords: Hybrid Ontologies, Collaborative Ontology Engineering

1 Introduction

In this paper, we extend a method for reaching a consensus on a description of the world – or an approximation thereof - within a community of stakeholders. This method exploits the natural language communication between the stakeholders. Even though the method adopted is intended for ontology engineering; aimed at producing application-independent descriptions of the world for semantic interoperability between autonomously developed information systems, the ideas presented here are easily extrapolated to other domain in which modeling (and the agreements leading those models) are critical for a successful project. [16] observed communication and comprehension problems within projects with groups whose members had different (IT) backgrounds. It is this problem that we wish to address in this paper. The better the understanding within (and even across) communities, the more likely that the (ontology) project will be successful. Thus methods will need to take into account the social processes and means used by the community to reach those agreements. Since the most advanced means of communication between humans is natural language, it will be beneficial to exploit this natural language communication in the agreement processes.

Starting from an existing framework for collaborative ontology engineering that takes into account both formal and informal descriptions of concepts, which we will describe later on, we ask ourselves the following questions: 1) what is the nature of the meaning agreements (esp. across communities), 2) are there different levels of committing to these models and can these be exploited for driving agreement processes.

The paper is thus organized as follows: starting from a brief introduction to ontologies, ontology engineering and related work, we move to the method in Section 3. Section 3 starts with a description from the hybrid ontology engineering framework and method we adopted in this paper, which is based on earlier work. In Section 3, we also describe how the nature of agreements across communities and propose to make a distinction between two types of ontological commitment: at community level, and at the level of a specific application. Ensuring proper business – or proper semantic interoperation – will be the motivation of this separation. We furthermore explain how the commitments can be used to drive the social interaction within the community that will lead to agreements. Section 4 presents the tools implementing these ideas and we conclude this paper in Section 5.

2 Related Work

An ontology is commonly defined as a formal, explicit specification of a shared conceptualization and ontology engineering is a set of tasks related to the development of ontologies for a particular domain. The semantics of an ontology stem not from the ontology language in which the ontology is implemented[1], but from the *agreements* of a community of stakeholders with a particular goal. Those agreements are achieved by interactions within the community leading the ontology to better approximate the domain over time.

We stated what ontologies are. The problem, however, is not what ontologies are, but how they become *community-grounded* resources of semantics, and at the same time how they are made operationally relevant and sustainable over longer periods of time. Quite a few surveys on the state of the art on ontology engineering methods exist [7,16,17]. Some collaborative methods provide tool support such as HCOME [9], DILIGENT [20] and Business Semantics Management[2] [2]. There even exists collaborative ontology engineering platforms, such as, Collaborative Protégé [18], that are not tailored to one specific method. Concerning methods, we noticed a between providing means for supporting social processes (in ontology engineering) and a special linguistic resource to aid these processes [4]. This gap was addressed in [3], which provided a framework for hybrid ontology engineering. Then, a method and tool were developed on top of this method, called GOSPL [4], which stands for Grounding Ontologies with Social Processes and Natural Language.

3 Method

In conceptual modeling, the natural language aspect helps us to keep a close communication link between the distinct stakeholders and the systems and/or business specifica-

---

[1]Although some constructs can be reserved a special meaning used for inference, e.g., the relation denoting subsumption

[2]http://www.collibra.com/products-and-solutions/products/business-semantics-glossary

tions. This has already been shown before in database design methods and techniques such as NIAM [21], which allows users to model their world by means of fact-types[3] expressed in natural language. In this section, we explain how we adopted fact-orientation for ontology engineering and use distinct levels of "precision" for describing concepts, informal and formal, with the formal level also being grounded in natural language This hybrid aspect is useful since we need informal descriptions to support high level reasoning among humans (i.e. discussions) and at the same time, formal descriptions to be used by machines.

3.1 A Framework and Method for Hybrid Ontology Engineering

Whenever two or more autonomously developed information systems need to interoperate, agreements over the concepts implicitly shared by those systems are made explicit, allowing the mapping of the conceptual schemas onto an ontology. Agreement processes thus co-exist at an organizational level and across organizations. The construction of an ontology can be supported by the same natural language fact-oriented modeling techniques. In fact, a framework for fact-oriented ontology engineering was proposed in [12] that adopted NIAM. This method was extended to include a special linguistic resource, called a glossary, to support the social processes in ontology engineering [3]. The social processes result in changes in the ontology and have been parameterized with the community, thus resulting in a well-defined hybrid aspect on ontologies. A Hybrid Ontology Description [3] contains:

- A lexon base Λ, i.e. a finite set of lexons. A lexon is a binary fact-type that can be read in two directions: t_1 playing the role of r_1 on t_2 and t_2 playing the role of r_2 on t_1 in some community referred to by $\gamma \in \Gamma$, where $t_1, t_2 \in T$ are term-labels and $r_1, r_2 \in R$ are role-labels. Communities are used to disambiguate agreements. An example of a lexon is *<Ticket Community, Ticket, has, of, Price>*.
- A glossary G, a finite set of functions mapping lexon or terms in lexons to natural language descriptions. For instance, the *Ticket Community* can agree to *articulate* the term *Price* with the gloss *"The sum or amount of money or its equivalent for which anything is bought, sold, or offered for sale."* The functions g_1 and g_2 map respectively community-term pairs and lexons to glosses.
- $ci: \Gamma \times T \rightarrow C$ a partial function mapping pairs of community-identifiers and terms to unique elements of C, a finite set of concepts.
- A finite set of ontological commitments K describing how one individual application commits to a selection of the lexon base, the use of this selection (constraints) and the mapping of application symbols to that selection. The elements of K will be described in the next section.

In [4], a collaborative method on top of aforementioned framework was described, called GOSPL. Fig. 1 depicts the processes in GOSPL. Communities define the semantic interoperability requirements, out of which a set of key terms is identified. Those terms need to be informally described before the formal description (in terms of lex-

---

[3] A fact-type is the generalization of facts, a collection of objects linked by a predicate. "[Person] knows [Person]" would be an example of a fact-type, and "[Christophe] knows [Cristian]" would be a fact in this example.

ons) can be added. In order for a lexon to be entered, at least one of the terms needs to be articulated. The terms and roles in lexons can be constrained. The community can then commit to the hybrid ontology by annotating an individual application symbols with a constrained subset of the lexons. At the same time, communities can interact to agree on the equivalence of glosses and the synonymy of terms. Important here is that the community first needs to "align" their thoughts and ideas by means of the informal descriptions before formally describing the concepts. This aids in avoiding misunderstandings and changes on the formal descriptions are then less likely to occur.

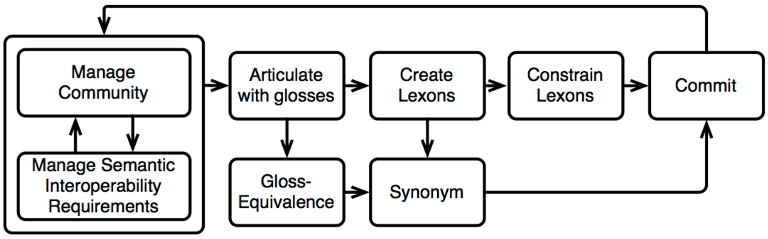


Fig. 1 The GOSPL method.

Important in GOSPL is that each "phase" corresponds with a number of social processes within a community. These social processes are there to discuss whether changes will contribute the community in achieving their goal. Rather than immediately change the ontology and discuss the change, the community needs to approve the proposed changes. Only when changes are accepted, they are carried out on the ontology. As the social processes are described and stored, we have added an additional dimension to traceability; the discussion and decisions made by the community.

3.2 The Nature of Agreements

Communities can agree that glosses used to describe terms can refer to the same concept as well as terms in lexons, gloss-equivalence (at gloss-level EQ_G) and synonymy respectively (at lexon-level $\equiv_C$). The elements in C contain the agreements of communities that a particular label refers – for all the members of a community – to the same concept. Every community-term pair refers to at most one concept, otherwise the community would be divided. Communities can agree that their terms could refer to the same concept and those agreements are captured. Given two community-term pairs $(\gamma_1, t_1), (\gamma_2, t_2) \in \Gamma \times T$, $ci(\gamma_1, t_1) \equiv_C ci(\gamma_2, t_2)$ denotes the agreement between communities γ_1 and γ_2 that their terms t_1 and t_2 refer to the same concept. The function g_1 maps every community-term pair to at most one gloss. Given communities $\gamma_1, \gamma_2 \in \Gamma$ and terms $t_1, t_2 \in T$, we say that two term-glosses $g_1(\gamma_1, t_1)$ and $g_1(\gamma_2, t_2)$ are gloss-equivalent EQ_G if the two communities agree that the described terms refer to the same abstract concept. A hybrid ontology is glossary-consistent if for every two pairs $(\gamma_1, t_1), (\gamma_2, t_2) \in \Gamma \times T : EQ_G\big(g_1(\gamma_1, t_1), g_1(\gamma_2, t_2)\big) \to ci(\gamma_1, t_1) \equiv_C ci(\gamma_2, t_2)$. The converse does not necessarily hold.

Note that when the two communities agree that the glosses used to describe their terms are gloss-equivalent, that this does not automatically imply that $ci(\gamma_1, t_1) \equiv_C ci(\gamma_2, t_2)$ is asserted. We motivate the reason to have both agreements established separately as follows: Gloss-equivalences are on the level of the glossary whereas $\equiv_C$ resides on the formal descriptions of the concepts (i.e. the lexons). To assert $\equiv_C$, the term must appear in a lexon. Communities can start gradually building their glossary before formally describing their concepts. However, nothing should prevent the community for having agreements on the "sameness" of descriptions across or within their own community. Another reason is validation of the equivalences. The glossary-consistency principle will pinpoint the descriptions used for terms that are EQ_G, but whose terms in those communities are not $\equiv_C$ The glossary-consistency principle does not become a property that needs to hold or else the ontology project fail, instead it becomes a tool to drive the community in establishing $\equiv_C$, double checking whether the gloss-equivalence was not misleading and both terms really do refer to the same concept.

This is particularly handy as the validity of the natural language descriptions and the equivalence of two such descriptions are relative to the communities partaking in these discussions. If glosses have been ill defined, yet agreed upon, the second agreement while the terms are formally described are more than welcome and the community will be able to rectify the mistakes.

Important to note is that assertions of gloss-equivalences and synonymy are only symmetric, reflexive and transitive *within one agreement process*. This measure is taken to avoid unwanted synonymy and gloss-equivalences to be propagated across communities. If communities A, B and CA all get together and agree that their terms tA, tB and tC are synonymous, the following assertions are added: $ci(A, tA) \equiv_C ci(B, tB)$, $ci(B, tB) \equiv_C ci(C, tC)$ and $ci(A, tA) \equiv_C ci(C, tC)$. However, if community C and D afterwards agree that $ci(C, tC) \equiv_C ci(D, tD)$, then this does not imply that $ci(A, tA) \equiv_C ci(D, tD)$ or $ci(B, tB) \equiv_C ci(D, tD)$. The agreements on synonymy can be followed will be followed by the other communities, allowing them to start interactions to state the terms are indeed synonymous. The same holds for gloss-equivalences.

3.3 Community- and Application Commitments

In GOSPL, a finite set of ontological commitments K contain descriptions on how individual applications commit to a selection of the lexon base (with constraints and mappings). We feel, however, the need to make a distinction between two types of commitments: community-commitments and application-commitments. The first is an engagement of the community members to commit to the lexons and constraints agreed upon by the community. The latter is a selection of lexons that are constrained (according to how the application uses these lexons) and a set of mappings from application symbols to terms and roles in that selection.

The introduction of a community commitment is motivated by the need for proper semantic interoperation between information systems. Depending on the goal of the ontology, instances shared across different autonomous information systems need to some degree to be compared for equivalence. One example is joining information about an instance across heterogeneous sources. In order to achieve this, the members of the community have to agree upon a series of attributes that uniquely, and totally identify the concepts they share. In other words, the conceptual reference structures[4]. By sharing the same reference structures, the information systems are able to interpret information describing instances and find the corresponding instance in their data store

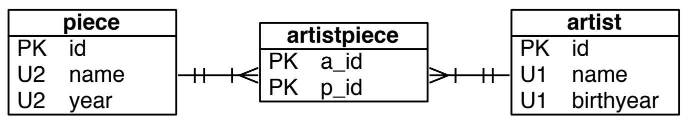
[4] Similar to identifications schemes in databases.

(of that of a third system). Application commitments refer to community commitments and can contain additional lexons and constraints. For instance, lexons needed to annotate application specific symbols (e.g., artificial IDs, often found in relational databases) to ensure that instances of concepts are properly aligned (e.g., a proper annotation of the foreign keys in a join-table). Both community- and application commitments also store information about the agreements across communities.

The application-commitment language we have adopted is Ω-RIDL [19], and extended to include references to community commitments. Take for example the ER-diagram for a fictitious database storing information about artists and works of art in Fig. 2. The corresponding application commitment is shown in Fig. 2. Notice the reference to the "Cultural Domain" community, which will include all lexons and constraints currently agreed upon by that community. This particular commitment furthermore includes some application specific knowledge to annotate the artificial IDs. The commitment describes how these IDs uniquely and totally identify instances of artists and works of art. Furthermore the terms "Artist" and "Work Of Art" inside the application's lexons are declared to be synonymous with that of the community. The lexons of the community 'Cultural Domain' g in this example were assumed to include:

```
<g, Art Movement, with, of, Name>         <g, Gender, with, of, Code>
<g, Artist, with, of, Art Movement>       <g, Artist, having, of, Name>
<g, Artist, born in, of birth of, Year>   EACH Name IS LEXICAL.
<g, Work Of Art, with, of, Title>         EACH Code IS LEXICAL.
<g, Work Of Art, made in, of, Year>       EACH Year IS LEXICAL.
<g, Artist, with, of, Gender>             EACH Title IS LEXICAL.
<g, Artist, contributed to, with contributor, Work Of Art>
```

The lexical constraints limit instances of concepts denoted by a term to "things" that can be printed on a screen.




```
BEGIN SELECTION
 # Selection of the community.
['Cultural Domain']
 # Application specific lexons
 <'MyOrganization', Artist, with, of, AID>
 <'MyOrganization', Work Of Art, with, of, WID>
END SELECTION
BEGIN CONSTRAINTS
 # Declaration of synonyms
 LINK('Cultural Domain', Artist, 'MyOrganization', Artist).
 LINK('Cultural Domain', Work Of Art, 'MyOrganization', Work Of Art).
 # List application specific constraints
 EACH Artist with AT MOST 1 AID.        #(1)
 EACH Artist with AT LEAST 1 AID.       #(2)
 EACH AID of AT MOST 1 Artist.          #(3)
 EACH Work Of Art with AT MOST 1 WID.   #(4)
 EACH Work Of Art with AT LEAST 1 WID.  #(5)
```

```
 EACH WID of AT MOST 1 Work Of Art.     #(6)
END CONSTRAINTS
BEGIN MAPPINGS
 # Mapping of application symbols, in this case from Table X Field
 # -> Term role Term (role Term)+, a path of lexons
 MAP 'Artist'.'name' ON Name of Artist.
 MAP 'Artist'.'birthyear' ON Year of birth of Artist.
 MAP 'Artist'.'id' ON AID of Artist.
 MAP 'piece'.'name' ON Title of Work Of Art.
 MAP 'piece'.'year' ON Year of Work Of Art.
 MAP 'piece'.'id' ON WID of Work Of Art.
 MAP 'artistpiece'.'a_id' ON  AID of Artist contributed to Work Of Art.
 MAP 'artistpiece'.'p_id' ON WID of Work Of Art with contributor Artist.
END MAPPINGS
```

Fig. 2 Example ER diagram and corresponding Ω-RIDL application-commitment

3.4 Exploiting Application Commitments

The application commitments – next to describing how the application symbols are related to the shared lexons – are useful for practical things such as: 1) the publishing of data in other formalisms and 2) the validation of one applications' data with respect to the concepts and constraints agreed upon by the community.

The GOSPL hybrid ontology engineering aims to facilitate the engineering of ontologies and the reduction of a knowledge engineer's involvement in the processes, diminishing the effort spent by experts. The framework aims to be ontology language agnostic. The grounding in natural language and restricting the knowledge building blocks to fact-types instead of making a distinction between classes and properties (or entities and relations) and having those fact-types expressed in natural languages leverages the modeling task. Hybrid ontologies are easily transformed into other formalisms and can be used in conjunction with those other formalisms currently used within semantic technologies. For instance, the ontologies are transformed into the Web Ontology Language (OWL 2 [8]) and used with the R2RML language [1] to offer a virtual SPARQL [13] endpoint over the mapped relational data, or generate RDF [10] dumps, or offer a Linked Data[5] interface.

Not only can hybrid ontologies by transformed to other formalisms, the application commitments also aid the transformation of data locked in closed information systems. [19] even described how mappings can be used to generate SQL queries for relational databases. Another implementation of Ω-RIDL - provided by Collibra NV/SA[6] - allow also the annotation of XML. No matter the formalism, a link with the hybrid ontologies is kept. This link allows exploiting the annotation to see to what extent the individual application comply with the constraints agreed upon by the community as well as those that are application specific. Transforming each constraint into a query does this.

The application mappings are changed according to each closer approximation of the observed world by the communities. As the hybrid ontology grows, so will the data unlocked by means of these commitments. In GOSPL, the constraints that are currently proposed in the community commitment can be tested against the data inside the

[5] http://www.linkeddata.org/
[6] http://www.collibra.com/

closed information systems in the same way that the constraints inside a commitment can be tested.

Also the hybrid ontologies and the additional lexons and constraints in application commitments are examined with a reasoner. This is particularly important for application commitments, as the annotations – being the responsibility of the representatives of that particular application – are human and thus inconsistencies could arise.

3.5 Queries as Concept Definitions

Lexons in a community commitment (or even an application commitment) can be used to query information by means of sentences created by concatenating lexons. We created a fact-oriented query language for RDF - called R-RIDL. For this, we adopted the fact-oriented query language RIDL [11]. RIDL, which stands for Reference and IDea Language, was a formal syntactic support for information and process analysis, semantic specification, constraint definition and a query/update language at a conceptual level in the early eighties. The RIDL language manipulated, defined and restricted information structures and flows described using the NIAM method (restricted to binary relations). RIDL was one of the first query languages to access the data via the conceptualization, which resulted from a natural language discourse between the users (of an information system). Because of its groundings in natural language, it was easier for users to retrieve information out of the system. A guide and description of the RIDL grammar are described in [6].

RIDL is a Controlled Natural Languages (CLN), which are less expressive subsets of natural language whose grammars and lexicons have been restricted, making it less complex and ambiguous [15]. CLNs make information retrieval and ontology engineering tasks easier on the user by hiding some of the complexity (e.g., learning standards such as XML, RDF and OWL) [15]. RIDL also inspired Ω-RIDL. Using a concatenation of lexons, sentences can be constructed to describe those application symbols.

Statements entered by the user are parsed following a grammar based on the original RIDL language; the part concerned with information retrieval and refined to cope with Hybrid Ontology Descriptions[7]. Below, we will give two examples of queries in R-RIDL with their equivalent expression in SPARQL. For the queries in SPARQL, the OWL translation of the community commitment is assumed to be available somewhere[8]. We omit the namespaces for the SPARQL queries for simplicity's sake. We assume the prefix of the OWL implementation of the community commitment to be `myOnto0`. Using the same example as the previous section:

- Return the artists that are not male

```
R-RIDL: LIST Artist NOT with Gender with Code = 'M'
```

```
SPARQL:  SELECT DISTINCT ?a WHERE { ?a a myOnto0:Artist.
          OPTIONAL { ?g myOnto0:Gender_of_Artist ?a.
                     ?g myOnto0:Gender_with_Code ?c. }
```

[7] Details of R-RIDL can be found on http://starlab.vub.ac.be/website/node/756/edit

[8] For this example, the OWL translation can be found on http://starlab.vub.ac.be/staff/chrdebru/GOSPL_ATOMIZER/art.owl

```
FILTER(?c != "M" || !bound(?c)) }
```

In this example we wish to list all the artists not having a gender with code `M'. This includes the artists whose gender was not explicitly stated. For the equivalent SPARQL query, we thus need to specify that gender is optional. This is done with the `OPTIONAL` clause, which will leave the variables unbound if no such information is available. But merely testing the whether variable `?c` doesn't equal `M' does not suffice. As apart from `bound`, all functions and operators that operate on RDF will produce a type error if any arguments are unbound. Thus the result of a Boolean test can be true, false or error. Testing whether `?c != `M'` will thus result in an error and the result will thus not taken into account for this query. We therefore need to test whether the variable doesn't equal `M' *or* the variable is unbound.

- Return all the names:

R-RIDL:
```
LIST Name.
```

SPARQL:
```
SELECT DISTINCT ?n WHERE {
  {?a myOnto0:Artist_having_Name ?n.} UNION
  {?a myOnto0:Art_Movement_with_Name ?n.}}
```

In R-RIDL, if we want to have the set of all names, we merely need to use that term label. This is not possible in SPARQL as lexical attributes result in object properties with their ranges being instances of `rdfs:Literal`. To achieve the same effect, i) one needs to look up all the lexons in which that term plays a role, ii) find the corresponding data properties and iii) construct the SPARQL query using the `UNION` operator for each of those data properties.

There are two types of statement: LIST and FOR-LIST. The LIST statement returns a set of instances, which can be regarded as a set of unary tuples. The FOR-LIST statement allows the user to create queries returning a set of tuples of arity n>1.

R-RIDL transforms parts of the lexon-paths in these queries into SPARQL queries, and then applies relation algebra to construct the result set. The drawback of this approach is that queries in R-RIDL are indeed slower than in SPARQL, but the added value is an understandable - and at certain points more expressive - query language for RDF fitting an ontology engineering method.. Where SPARQL is suitable for building services, R-RIDL allows for language-grounded exploration of data.

GOSPL allows agreements to be made at two levels: at description level and at the level of the formalism (i.e., lexons). Even though the method supports both high level reasoning by humans with the natural language descriptions and low level reasoning by machines with the formal part, the ontology engineering processes can benefit from the hybrid nature of R-RIDL; the queries looking like natural language sentences become concept definitions. The definition/query can be defined, as the results can be explored, examined and discussed by the community. Those definitions correspond with the *subtype definitions* of ORM, in which subtypes of concepts are defined in terms of the roles of lexons played by its super-types. For instance:

- `EACH Female Artist IS IN` **`LIST Artist with Gender with Code = 'F'`**, Or
- `EACH Female Artist is a` **`Artist with Gender with Code = 'F'`**

4 Tool and Demonstration

The following principles have been included in a tool called GOSPL [5,4]. Fig. 3 depicts a screenshot of the GOSPL prototype, and shows some lexons and constraints currently residing in the "Venue Community", which aimed to describe the venues in which cultural events take place. The tabs in this figure direct the user to:

- **Ontology.** The lexons and constraints currently agreed upon by the community. This is actually the community-commitment
- **Glossary.** The natural language descriptions for terms and lexon currently agreed upon by the community. This page also displays the current gloss-equivalences and to what extent the hybrid ontology is glossary-consistent.
- **Discussions.** The social processes as discussions to evolve the hybrid ontology, as well as the semantic interoperability requirements.
- **Members.** Community management. We choose not to assigned roles denoting a hierarchy; instead we choose to treat all members equal. This simplifies teaching the method.
- **Commitment.** The list of application-commitments. Such commitments can exist without the platform knowing about its existence. However, for the system to be able to query data or test constraints proposed by the community, the systems needs to keep track of applications whose application symbols are annotated. Users are able to manage application commitments expressed in Ω-RIDL and SPARQL-endpoints, with the latter preferably providing triples using predicates from the OWL implementation of the hybrid ontology.
- **OWL/RDFS.** The OWL implementation of the hybrid ontology
- **Activity.** A log of this particular community

Fig. 4 depicts a simple "scenario" with the tool. After logging in, users a presented a list of communities (A), users can take a look in each community – for instance the Venue community in (B) and the discussions of that community (C). The image in (B) corresponds with the screenshot in Fig. 3. Depending whether the user is a member of a community, the user has access to a number of social processes he can start within that community. In (D), we show how a discussion to add a gloss is started. The discussion presented in (E) stems from the experiment we will describe later on. Once a term is articulated, lexons can be built around this term (F) and constraints on the created lexons (G). After a while, the community has obtained a closer approximation of their domain and can start creating/updating their application-commitments (H). These commitments can be (users are not obliged) registered to the platform, which can then be used to test statements made in a discussion, e.g., by looking for counter-examples (H). When users are not part of a community, the interactions they can start only involves general requests (e.g., request an edit, or request to become a member), they have no access to requests on the glossary or lexon base. If that user is part of another community, he can trigger processes to discuss the "sameness" of glosses or terms.

Information on synonymy and gloss-equivalences are shown on a separate page (a community-term page), accessible by - for instance - clicking on one of the terms of the accepted lexons. The GOSPL tool supports a community in *applying* the method for ontology engineering, but its purpose is indeed not to replace other means of interaction that can be more effective when possible (e.g., face-to-face meetings when community members are near, or even teleconferences). The outcome of these interactions outside of the tool, however, needs to be properly written down when concluding a discussion.

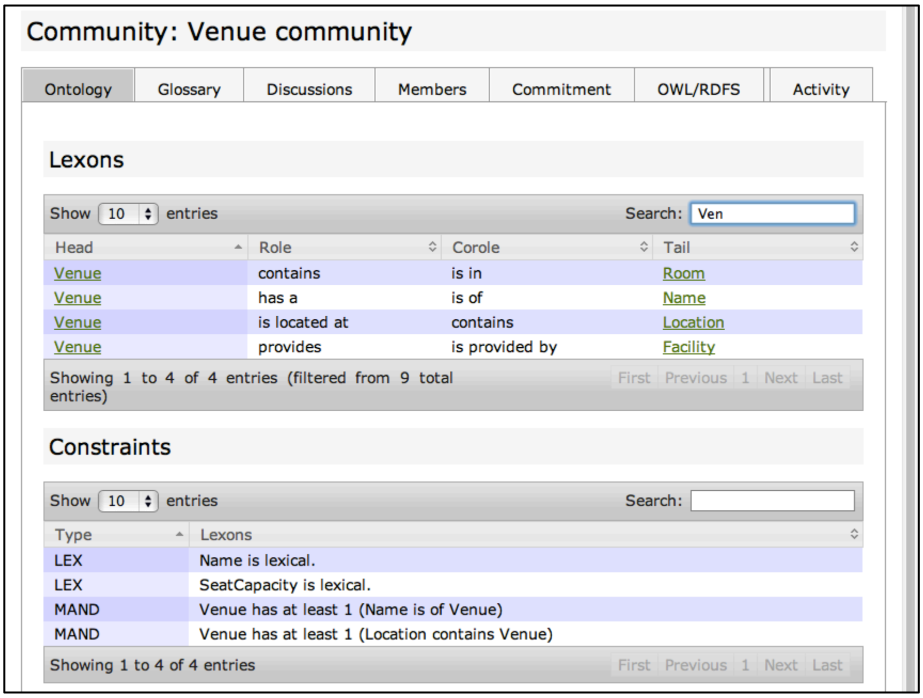


Fig. 3 Screenshot lexons and constraints in a community

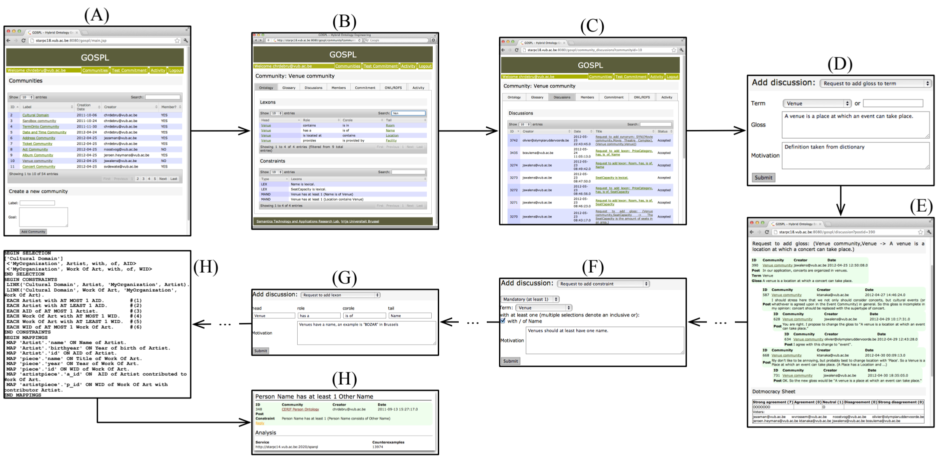


Fig. 4 Different social processes supported by the tool.

In the previous section, we described how the lexons in the hybrid ontology-engineering framework could be used to create controlled natural language queries.

Given a commitment and the SPARQL end-point, the first tells the client to which community-commitments this application is committing two and the latter where the data is available. Application-commitments can be used to generate mapping files, e.g. with R2RML, but details this will be reported elsewhere. In short, the Ω-RIDL annotations are analyzed to construct the appropriate mappings, taking one special case into account: whether join-tables in the relational database are represented by a lexon in the hybrid ontology, or as a term. Whether the mapping is generated, or done manually, R-RIDL is able to return results when data is annotated with the hybrid ontology.

GOSPL provides the knowledge management platform for managing and creating the ontologies for a Linked Data project in Brussels. One of the use cases is the publication of information related to cultural events taking place in Brussels. To this end, we conducted an experiment with 41 volunteers, each divided in subgroups of 3 to 4 people. Each group was asked to come up with an application in the domain of cultural events and then to create a hybrid ontology to enable semantic interoperability between their systems and one provided by the use case partners. This experiment lasted 7-8 weeks, and several communities were created. The groups had a natural tendency towards separating concerns, creating communities that complemented each other. For instance, the creation of a "Ticket" community for a general description of tickets, conditions and prices. We analyzed the interactions involving terms in a community with the following criteria: (1) The term had to be non-lexical, meaning that instances of this concept cannot be printed on a screen, only it's lexical attributes can. (2) The term was the subject of at least 4 interactions (not including gloss-equivalences and synonyms, thus focusing on the formal and informal descriptions around this term). (3) The term took part in at least one lexon.

We took into account terms with a fair amount of activity. This is due to the fact that the communities employed terms only relevant to their application, and therefore only inspired discussions within that group. These discussions are not interesting as the community tended to agree on what has been decided for their application.

We then analyzed how much of these terms changed in terms of their formal description if a gloss was immediately provided. With these criteria, we identified 49 terms. Of these 49 terms, 38 started with the natural language description as described by the GOSPL method. Of these 38 terms, 11 of them had changes in their formal description (29%). And of the remaining 11 terms that did not start with the informal description, 5 of them changes in their formal description (45%).

The reason we left out lexicals is that they often play in an attributive role. Lexons are supposed to be entered when at least one of the terms is informally described. At the start, the key-terms are often described first. And when the second term concerns a lexical in an attributive role, the community tends to agree on the meaning of this attribute based on the label of that term. If we were to take lexicals into account, we again observe that terms that did not start with an informal description are more likely to change its formal description: 18 terms out of 46 that started with a gloss and 6 terms out of 12 that did not start with a gloss.

5 Conclusions

In any project in which agreements within a heterogeneous community of stakeholders are vital, the natural language aspects in communicating knowledge and aligning ideas are key for success. In this paper, we described – in the context of ontology engineering - how agreements within and across communities are facilitated by natural language descriptions. The ideas presented in this paper are easily extrapolated to other domain, e.g., large software projects, in which the construction of a lexicon for use between developers, users, and other stakeholders will be used throughout the project.

We introduced the notions of community- and application commitments. The first captures the agreements by one community necessary to achieve the community's goals, the latter to ensure proper interoperation by one application. Application commitments even provide additional information if the owners of that application wishes to. This layered approach is also easily applicable in different domain, where the community commitment will contain fact-types and business rules that should always hold and application-commitments contain additional fact-types and rules for specific application (e.g., the rules to which an instance of a concept must comply with in different stages of that entities lifecycle management). We described the nature of agreements; the "sameness" of term-labels or glosses is considered an equivalence relation only within the communities participating in one agreement process.

We furthermore described how these application commitments aid the ontology engineering processes in guiding the interactions within the community. Hypotheses are transformed into queries that returning instances that do not support the hypothesis. The application commitments that co-evolve with the community commitments, allow 1) to publish information in those applications as structured data on the web and 2) users to also explore already annotated data and examine any other annotation on these instances (not necessarily with knowledge from the community). The latter is done by means of R-RIDL, a fact-oriented query language on top of RDF. R-RIDL is a controlled natural language using the natural language fact-types agreed upon by the community. Expressions in R-RIDL allow describing how instances are classified by means of a query, much like subtype-definitions used in ORM.

We implemented these concepts in a tool for hybrid ontology engineering, called GOSPL, and conducted an experiment. One problem we encountered was a tendency by the communities to forget describing the lexical terms with a natural language description. However, it is important – for some concepts – to agree on how some lexical entities should be represented (in terms of format, encoding, etc.). The tool should thus be altered in such a way that communities are still encouraged to describe all of the term in an informal way, even when they're "merely" lexical attributes. The prototype was developed with respect to the method. Some freedom, however, was granted to the users; e.g., terms did not have to be articulated for lexons to be created around it. In a next experiment, we will impose this constraint and examine the users' reactions on this change. At the same time, we will investigate how we should put emphasis on this issue while teaching the method to the participants.

Acknowledgements. This work was partially funded by the Brussels Institute for Research and Innovation through the Open Semantic Cloud for Brussels Project.

References

1. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. W3C Working Draft 29 May 2012 (2012) via http://www.w3.org/TR/r2rml/
2. De Leenheer, P., Christiaens, S., Meersman, R.: Business semantics management: A case study for competency-centric HRM. Computers in Industry 61(8) (2010) pp. 760–775

3. Debruyne, C., Meersman, R.: Semantic interoperation of information systems by evolving ontologies through formalized social processes. In: ADBIS: Eder, J., Bielikova, M., Tjoa, A.M. (eds.). Vol. 6909 of LNCS, Springer (2011) pp. 444--459
4. Debruyne, C. and Meersman, R.: GOSPL: a Method and Tool for Fact-oriented Hybrid Ontology Engineering. In Proc: ADBIS: Haerder, T., Morzy, T., Wrembel, R. (eds.). Vol. 7503 of LNCS, Springer, (2012) pp. 153-166
5. Debruyne, C., Reul, Q., and Meersman, R.: GOSPL: Grounding Ontologies with Social Processes and Natural Language. In: ITNG 2010: Latifi, S. (eds.), IEEE Computer Society (2012) pp. 1255-1256
6. De Troyer, O., Meersman, R., Ponsaert, F.: RIDL user guide. Research report (available from the authors), Int. Centre for Information Analysis Services, Control Data (1983)
7. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2003) ISBN: 1852335513.
8. Hitzler, P., Krotzsch, M., Parsia, B., Patel-Schneider, P. F., Rudolph, S.: OWL 2 Web Ontology Primer. W3C Recommendation 27 October 2009 (2009) via http://www.w3.org/TR/owl2-primer/
9. Kotis, K., Vouros, A.: Human-centered ontology engineering: The hcome methodology. Knowledge Information Systems 10(1) (2006) pp. 109–131
10. Manola, F., Miller, E.: RDF Primer. W3C Recommendation 10 February 2004 (2004) via http://www.w3.org/TR/rdf-primer/
11. Meersman, R.: The RIDL conceptual language. Research report (available from the authors), Int. Centre for Information Analysis Services, Control Data (1982)
12. Meersman, R.: The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In: The Proceedings of the Second Int. Symposium on Cooperative Database Systems for Advanced Applications (CODAS99): Zhang, Y., Rusinkiewicz, M., Kambayashi, Y. (eds.), Springer (1999) pp. 1--14
13. Prud'hommeaux, E., Seaborne, A., SPARQL Query Language for RDF. W3C Recommendation 15 January 2008 (2008) via http://www.w3.org/TR/rdf-sparql-query/
14. Sahoo, S. S., Halb, W., Hellmann, S., Idehen, K., Thibodeau, T., Auer, S., Sequeda, J. and Ezzat, A.: A survey of current approaches for mapping of relational databases to RDF (2009) via http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf
15. Schwitter, R.: A controlled natural language layer for the semantic web. In: Australian Conference on Artificial Intelligence: Zhang, S., Jarvis, R. (eds.). Vol. 3809 of LNCS., Springer (2005) pp. 425—434
16. Simperl, E., Tempich, C.: Ontology engineering: A reality check. In: OTM Conferences (1): Meersman, R., Tari, Z. (eds.). Vol. 4275 of LNCS, Springer (2006) pp. 836–854
17. Siorpaes, K., Simperl, E.: Human intelligence in the process of semantic content creation. World Wide Web 13(1-2) (2010) pp. 33–59
18. Tudorache, T., Noy, N., Tu, S., Musen, M.: Supporting collaborative ontology development in protégé. In: International Semantic Web Conference: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.). Vol. 5318 of LNCS., Springer (2008) pp. 17–32
19. Verheyden, P., De~Bo, J., Meersman, R.: Semantically unlocking database content through ontology-based mediation. In: SWDB: Bussler, C., Tannen, V., Fundulaki, I. (eds.). Vol. 3372. (2004) pp. 109—126
20. Vrandecic, D., Pinto, H.S., Sure, Y., Tempich, C.: The DILIGENT knowledge processes. Journal of Knowledge Management 9(5) (2005) pp. 85–96
21. Wintraecken, J.: The NIAM Information Analysis Method, Theory and Practice. Kluwer Academic Publishers (1990)