Christophe Debruyne and Robert Meersman. GOSPL: A Method and Tool for Fact-Oriented Hybrid Ontology Engineering. In T. Morzy, T. Härder, and R. Wrembel, editors, ADBIS, volume 7503 of LNCS, pages 153-166. Springer, 2012

GOSPL: a Method and Tool for Fact-oriented Hybrid Ontology Engineering

Christophe Debruyne and Robert Meersman

Semantics Technology and Applications Research Lab,
Vrije Universiteit Brussel, Brussels, Belgium
{chrdebru,meersman}@vub.ac.be

Abstract. In this paper we present GOSPL, which stands for Grounding Ontologies with Social Processes and Natural Language. GOSPL is a method and tool that supports stakeholders in iteratively interpreting and modeling their common hybrid ontologies using their own terminology for semantic interoperability between autonomously developed and maintained information systems. Hybrid ontologies are ontologies in which concepts are both formally and informally described with the help of a special linguistic resource called glossary. Social interactions between the community members drive the ontology evolution process and result in more stable and agreed upon ontologies.

Key words: Ontology Engineering Methodologies

1 Introduction

An ontology is commonly defined as: *"a [formal,] explicit specification of a [shared] conceptualization"* [10]. However, the problem 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. In the DOGMA framework [17], *fact-oriented approaches* such as NIAM/ORM [25,12] have been proven useful for engineering ontologies. A key characteristic here is that the analysis of information is based on natural language fact-types[1]. This brings the advantage that "layman" domain experts are facilitated in building, interpreting, and understanding attribute-free[2], hence semantically stable ontologies, using their own terminology. The semantics in ontologies are thus the result from *agreements* within a community.

An important tool in reaching agreements is the use of glosses, natural language descriptions interpretable by humans. The use of glosses while reasoning and discussing concepts among humans aid in the disambiguation of different

[1] 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 [Robert]" would be a fact in this example.

[2] There are only fact-types, no distinction between relations and attributes. The constraints on roles in these facts determine the "attributeness" of a fact-type.

concepts, discovery of implicit relations between concepts, discovery of gaps in the ontology, etc. We call the process of describing with a natural language description *articulation*. Ontology construction must be viewed as a complex, social and distinct methodological activity. It must lead to formalized semantic agreement involving its stakeholder communities and the various social processes within those communities. Enabling system interoperability therefore explicitly involves hybrid aspects of information; i.e. the co-existence of formal reasoning and "informal" human interactions (with natural language). In previous work [7], we presented a formalism and initial prototype for the engineering of so called hybrid ontologies. In hybrid ontologies, communities are promoted to first-class citizen, part and parcel of the formalism, such that the interactions within the evolving community result series of ontology evolution operators. The natural language aspect is vital, as the closer the link between human communication and the resulting system and/or business communication, the more likely such systems will work as intended by their various stakeholders.

The GOSPL method is a teachable and repeatable *collaborative ontology evolution method* supporting stakeholders in interpreting and modeling their common ontologies in their own terminology and context, and feeding back these results to the owning community. In this paper, we present the GOSPL method and prototype for fact-oriented hybrid ontology engineering built on top of the framework for hybrid ontology engineering described in [7].

2 Related Work

For the last twenty years, many methods have been put forward for how to develop ontologies. It seems, however, that research on methodological activities has diminished in recent years [1]. Bergman observed that very few discrete methods exist and are often older in nature [1]. He furthermore noted that most methodologies shared a number of logic steps from assessment to deployment, from testing to refinement. Quite a few surveys on the state of the art on ontology engineering methodologies exist [19,18,9]. From Table 1, we see that there is a gap between providing means for supporting social processes for elicitation and agreements and a special linguistic resource to aid these processes except - to our knowledge - for DOGMA [17]/GOSPL [7]. DOGMA already provided a framework for linking concepts with natural language definitions, which has been refined and augmented with social processes in [7].

Even though several (collaborative) ontology engineering tools exist such as NeOn[3], KAON2[4] and Collaborative Protégé [20] which uses underlying formalisms such as OWL-DL, SWRL and/or F-Logic, we restrict ourselves to describing the fact-oriented tools as we will adopt a fact-oriented method for ontology engineering. DogmaModeler[5] is an implementation of the DOGMA framework for creating, editing and representing ontologies, based on the separation

[3] `http://neon-toolkit.org/`
[4] `http://kaon2.semanticweb.org/`
[5] `http://starlab.vub.ac.be/website/node/47`

Table 1. Comparison of different ontology-engineering methodologies: Cyc [11] (A), DILIGENT [24] (B), DOGMA [17]/GOSPL [7] (C), HCOME [15] (D), [13] and [14] (E), METHONTOLOGY [8] (F), On-To-Knowledge (G), Unified Method [22,21] (H), and UPON [6] (I).

	A	B	C	D	E	F	G	H	I
Intended for collaboration	N	Y	Y	Y	Y	N	N	Y	N
NL description of concepts	Y[0]	N	Y	Y	N	Y	Y[0]	Y	Y
Special linguistic resource	N	N	Y	Y[1]	N	N	N	N	Y
Tool support	Y	Y[2]	Y	Y	N	Y	Y	N	N
Support for social processes	N	Y	Y	Y	N	N	Y	N	N
Social Processes for elicitation	N	N	Y	N	Y[3]	N	Y	Y	Y
Social Processes for agreements	N	Y	Y	Y	Y	N	N	N	N

[0]As meta-data or comments in the ontology. [1]Uses an external resource and users seem not able to introduce new descriptions. [2]Adopted an existing tool to support the processes. [3]Indirectly, as the knowledge is elicited at the same time as consensus is achieved on the different opinions/solutions generated by the community.

of the ontology base and ontological commitments. It was designed as a stand-alone application with no means for collaborative ontology engineering. However, it already provided an implementation of the DOGMA model into other formalisms. DOGMA-MESS [3] was a stand-alone fact-oriented ontology engineering tool for collaborative ontology engineering communicating with a server. Collaboration was achieved by means of tasks assigned to persons, called tickets. DOGMA-MESS also provided support for conflict management. Business Semantics Glossary (BSG)[6] is a commercial tool for collaborative fact-oriented knowledge modeling built on top of a wiki application. In BSG, the governance models are built in and user roles can be applied to distribute responsibilities. Due to its wiki-paradigm, however, users can deviate from the Business Semantics Management (BSM) [2] method. For goal driven communities, we need well defined processes and guidelines to achieve those goals.

3 Hybrid Ontology Engineering

In conceptual modeling, the natural language aspect is vital, as the closer the link between this human NL communication and the system and/or business communication that results from the system, the more likely such systems will work as intended by their various stakeholders. This has already been shown before in database design methods and techniques such as NIAM and ORM, which allows users to model their world by means of fact-types expressed in natural language.

Whenever two or more autonomously developed information systems need to interoperate, agreements over the concepts implicitly shared by those sys-

[6] `http://www.collibra.com/`

tems need to be made by the stakeholders in such a way that the conceptual schemas of those information systems can be mapped 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 [17] and was extended to include a special linguistic resource, called a glossary, to support the social processes in ontology engineering [7]. 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 [7] contains:

1. 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 context $\gamma \in \Gamma$, were $t_1, t_2 \in T$ are terms and $r_1, r_2 \in R$ are role-labels. In hybrid ontologies, contexts refer to communities in which agreements take place. An example of a lexon is ⟨ *Ticket Community, Ticket, has, of, Price* ⟩.
2. 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."*.
3. $ci : \Gamma \times T \to C$, a partial function mapping pairs of context identifiers and terms to (unique) elements of C, a finite given set of concepts.
4. 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 description of commitments falls outside the scope of this paper, for this we refer the reader to [23].

Communities can furthermore agree that glosses used to describe terms can refer to the same concept as well as terms in lexons, gloss-equivalence (at gloss-level) and synonymy respectively (at lexon-level). A motivation for this will be given in Section 4.5.

We defined a set of social processes on a community that are intended to reflect its member interaction with the "real world" and with each other and "map" those processes onto a sequence of ontology evolution operators. It is essential to observe that the ontology description evolves only as the result of agreements, viz. actions performed in principle by multiple community members. Some of those social processes are presented in Table 2. The glosses play a vital role, as they will facilitate agreements across communities and agreements on the formal descriptions of concepts.

GOSPL allows the community to have the ontology engineering process to be driven by the glosses, helping the community to start social processes to propose additional fact-types in the ontology. As discussions are started, the community as a whole can also control which discussions fall out of the scope. At some point, the community deems the Hybrid Ontology Description mature enough to start annotating their data and services. The Hybrid Ontology Description can then be implemented in another formalism such as OWL, a process that is done automatically. This OWL file can then be used to generate, amongst others, SPARQL endpoints that allows the data to be queried via the ontology. Even though we presented a framework in which this would be possible, the community still needs a method to achieve their goals. We will present this method in the next section.

4 GOSPL Method

In the previous section, we introduced a framework for hybrid ontology engineering on top of DOGMA, a fact-oriented ontology engineering approach. We furthermore defined several social processes that allow a community to alter the hybrid ontology towards a closer approximation of the community's domain. In this section, we present the method for hybrid ontology engineering. A method prescribes certain guidelines and steps to be taken to achieve a certain goal; in this paper, the construction of a hybrid ontology. The method uses the hybrid ontology-engineering framework and social processes defined in the previous section.

Fig. 1 summarizes the different processes in GOSPL. Starting from co-evolving communities and requirements, the informal descriptions of key terms have to be gathered before formally describing those concepts. These formal descriptions can be constrained and then committed to by application by using a commitment language, e.g., *Ω*-RIDL [23]. During the processes from creating the glossary to committing to the hybrid ontology description, the communities can make agreements on gloss-equivalences and synonyms. The hybrid ontology, and the data described with those commitments can then be re-internalized by the community for another iteration, gradually approximating the domain that needs to be captured by the ontology.

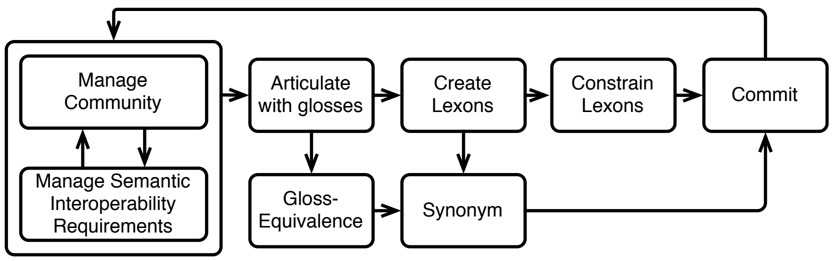

Fig. 1. The GOSPL method.

Table 2. Social processes in the GOSPL method.

Social process: Request to ...	
Add/Remove/Change gloss to/from/of lexon or term	Add/Remove lexon
Add/Remove gloss-equivalence	Add/Remove constraint
Change concept/role hierarchy	Add/Remove synonym

4.1 Defining Semantic Interoperability Requirements

We restrict ourselves to communities of users representing autonomously developed and maintained information systems with a need to exchange information for a purpose. This need is translated into a semantic interoperability requirement (SIR). The objectives of a SIR are to ensure the application or components interoperate with other specified information systems and their components. The data needs to be exchanged between those components and be useable upon reception and the different components “know” how to consult the data from other information systems or components. A community is thus partly identified by its SIRs. As we will see later on while describing the co-evolution between communities and their SIRs, we will identify communities by those requirements and its set of members.

A SIR for a community $\gamma \in \Gamma$ consists of a tuple $\langle KT, GO \rangle$: a non-empty set of key terms $KT \subset \Gamma \times T$ and a non-empty set of goals for which descriptions of those key concepts are needed. The community interacts and agrees upon the elements in those two sets.

4.2 Building the Glossary

Interoperability is achieved by annotating the symbols of an information system with terms and relations in the hybrid ontology. As the hybrid ontology and the glossary are initially empty, we must ask ourselves how these ontologies come to be. We have already described how in a hybrid ontology terms are on one hand described informally by means of natural language descriptions called glosses for humans and described formally for annotating information systems and their computerized systems on the other. To ensure all members of a community are referring to the same referent for a particular label, the community needs to align their ideas of the concept symbolized by the term. We call this process *alignment*. Alignment is achieved by (1) describing the concepts referred to by these labels and (2) having the community members agree on one such description.

To facilitate alignment, GOSPL imposes terms to be described informally before formal descriptions are added, staring with the list of key terms in the SIR.

4.3 The Creation of Lexons

Lexons can only be entered in the lexon base when one of the terms in this lexon has already been described informally. Indeed, it would be undesirable to describe a relation between two terms if both terms playing the roles in that relation are not described themselves, meaning that their intended meaning has not yet been made explicit. If at least one of the terms described, one can assume that the lexon proposed around that term is in function of the informal definition and/or the SIRs.

For the social process “Request to change superlexon of lexon (role hierarchy)”, however, we require that the four terms of both lexons involved be articulated. Indeed, how can one imply that an instance playing a particular role "r1" implies that same instance playing another role "r2" if the terms or the relation itself are not specified. Remember that lexons can be articulated as well only if both its terms are defined.

Lexons can be articulated with a gloss only if both its terms are articulated. In GOSPL, a community is able to articulate all the lexons. However, GOSPL strongly encourages articulating lexons whose internal uniqueness does not span only one role. In other words, GOSPL encourages the articulation of "many-to-many" relations in ER terminology. In the absence of an internal uniqueness constraint, the uniqueness constraint is assumed to be spanning the two roles. Such relations must correspond with a concept in the domain that needs to be approximated by the ontology. This is in contrast with so called "attributive" relations, which can be too "trivial" to fully articulate. Take for instance the lexon ⟨ C1, Person, working for, employing, Organization ⟩ where a person can work for many organizations and an organization can employ many employees. This many-to-many relation could denote the concept of position. In the example of ⟨ C2, Person, born on, of birth of, Date ⟩ with a person born on at most one Date, date (of birth of) becomes an attribute of Person. We therefore don't need to describe the relation as being the occurrence of persons having a birth date. Our claim is that non-attributive relations denote concepts, and therefore need to be described by the community. The relation between concepts and non-attributive relations will become apparent after we will treat the constraint one can put on lexons and reference structures of concepts.

4.4 Constraining Lexons

Commitments contain (1) a selection of lexons from the hybrid ontology, (2) mappings from application symbols to terms and roles in that selection and (3) constraints on that selection that indicate how that particular application uses those concepts. Some of these constraints have to be shared and agreed upon by the community in order to meet the interoperability requirements. Those constraints should not stem from the individual applications, but be part of the domain that has to be modeled. A classic example of such a constraint is book being uniquely, and totally identified by its ISBN number. Those constraints are needed to ensure proper interoperation between the different systems.

The community thus might need to agree on constraints in order to meet the goals captured by their SIRs. We make a distinction between two types of constraints: on terms and on roles of lexons. In either case, the GOSPL method imposes the terms to be articulated with a gloss. Indeed, it would be undesirable to constrain the use of a term, a role, or a lexon with insufficient articulation, as this means that their intended meaning has not yet been made explicit.

4.5 Gloss-equivalences and Synonyms

At any point in time, two communities can agree the glosses used to describe their terms are referring to the same concept, even if that term is not (yet) appearing in a lexon. This can be achieved with a social process to assert a gloss-equivalence between the two glosses. Note that there are two special cases of gloss equivalence: one in which the communities are different and the terms are the same (term-equivalence) and one in which the terms are different but the within the same community (community-equivalence).

Note that for an ontology to be consistent, if for every two community-term pair their glosses are identical or equivalent, there should be an agreement on the concept-identifiers. The inverse should not necessarily hold. Two concepts could be deemed synonyms by the communities, but their glosses not equivalent. This distinction is made to allow agreements to be made at the level of the glossary and at the level of the more formal lexons.

We motivate the agreements on these two levels. First, communities can start gradually building their glossary before formally describing their concepts. However, nothing should prevent the community for having agreements on the synonymy of concepts across or within their own community. If the definition would impose synonymy on the formal descriptions, the community first needs to agree on at least one lexon concerning that term. Another reason is validation of the equivalences. The glossary-consistency principle will pinpoint the descriptions used for terms that are gloss-equivalent, but whose terms in those communities are not synonymous. This principle will thus drive agreement processes.

4.6 Committing to the Hybrid Ontology

Once there is a close approximation of a (part of) the hybrid ontology for meeting the SIRs, the stakeholders can start annotating their information systems, with the hybrid ontology by means of a commitment. The commitments enable the exchange of information residing in those systems. With every (closer) approximation of the domain with the hybrid ontology, the commitments will provide access to instances of concepts that can be used for defining and/or refining the definitions, fact-types and constraints in the hybrid ontology description. How these instances are exploited for the definition and refinement of definitions, fact-types and constraints is, is part of our future work.

4.7 Community and SIR Co-evolution

We explained how a community starts the development of a hybrid ontology by first defining their SIRs, articulate the key terms in those requirements and gradually construct agreements on fact-types, glosses, constraints, gloss-equivalences and synonyms. Communities and their SIRs are, however, not static, they are evolving and even co-evolving. With the additional of a new stakeholder in the community, the community changed not only with the presence of a new member, but also with the addition of new ideas, a possible different perspective on matters and possible new requirements for the community. Also requirements can change from external forces, e.g. due to legislation changes. The community constitution does not necessarily need to change for the SIRs to evolve, a

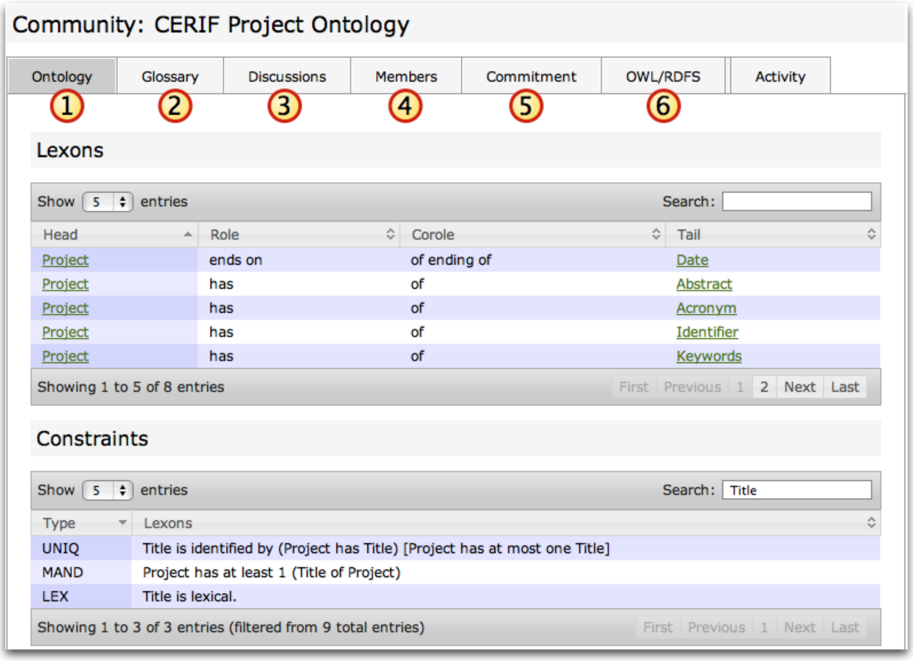

Fig. 2. Screenshot of the lexons and constraints of one communities' hybrid ontology description.

community can come to the conclusion that the current approximation of the domain by the hybrid ontology description does not meet their needs even though it complied with the requirements. In that case, the community will negotiate changes to the requirements. This can happen when the community starts to better understand the domain.

5 GOSPL Tool

The tool is developed in Java and runs inside an application container such as JBoss. It contains two layers: the base layer contains all the domain classes and communication with the server and a web application providing the interface layer. The base layer can also be consulted by other software agents, making the development of standalone clients possible. Fig. 2 shows a screenshot of the GOSPL tool. It shows a screen with the lexons and constraints of one communities' hybrid ontology description (1) and glossary (2), links to the discussions (3), community management (4), the commitments of applications to the ontology (5) and the OWL/RDF(S) implementation of the hybrid ontology (6).

GOSPL is discussion-oriented and both the ontology and glossary evolve only if the community reaches an agreement. This results in traceability also on decision level and not only at change level. In Fig. 3, one such (small) discussion is highlighted. Different discussion can be started. Depending whether a person is a member of the community, some discussions might not be available. However, all users can leave comments and all users can start "informal" discussions (even when they are not part of the community). In other words, we not only record who changes what, but also the reasons certain changes have been made by linking changes to discussion on the platform. This was possible by formalizing the social processes and its corresponding operators.

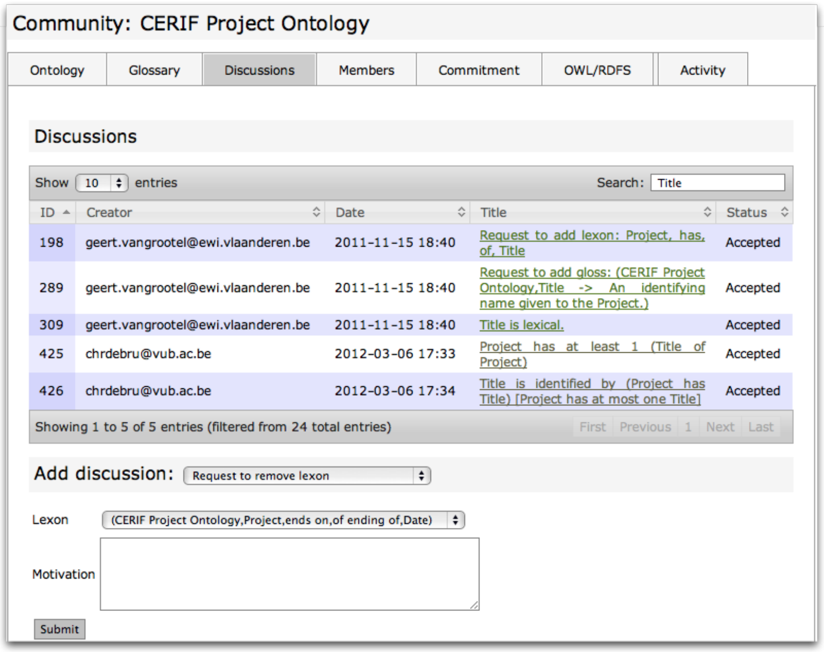

Fig. 3. Discussions (social processes) in GOSPL.

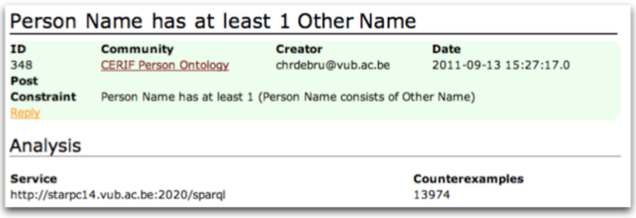

Fig. 4. Finding counterexamples for statements in the hybrid ontology.

A quasi-anonymous voting system is used to gather the opinion of people without the need of participating in the discussion. It is quasi-anonymous in the sense that anyone can see who has voted, but not what they voted on a scale from strong agree to strong disagree. This gives community member an idea who cares for particular parts of the ontology in a sufficiently large community.

The application commitments belonging to community members describe how the application symbols of their system commit to the ontology, allowing the information in those database systems to be retrieved through the ontology. Of course, the discovery of counterexamples does not necessarily mean that the statement is false, however, this information might direct the discussion into another direction. Fig. 4 shows a dataset has over 13000 counterexamples for the mandatory constraint on "has" between "Person" and "Other Name".

Fig. 5 depicts the description of community-term pair ⟨ CERIF Project Ontology, Project ⟩. GOSPL also shows the communities adopting this gloss or the glosses that the CERIF Project Ontology has adopted for this term. Glosses are a very important means to achieve consensus within and across communities. Others can easily start a discussion to state that this gloss is equivalent with another gloss (3). The application furthermore suggests the community members to introduce concepts, fact-types, etc. distilled from this gloss (2). Glosses thus provide "food for thought" to refine or complete the formal part of the hybrid ontology, a process that can be facilitated by the tool. This information can be then exploited to guide the discussion processes, by transforming certain statements into queries that will look for counter examples.

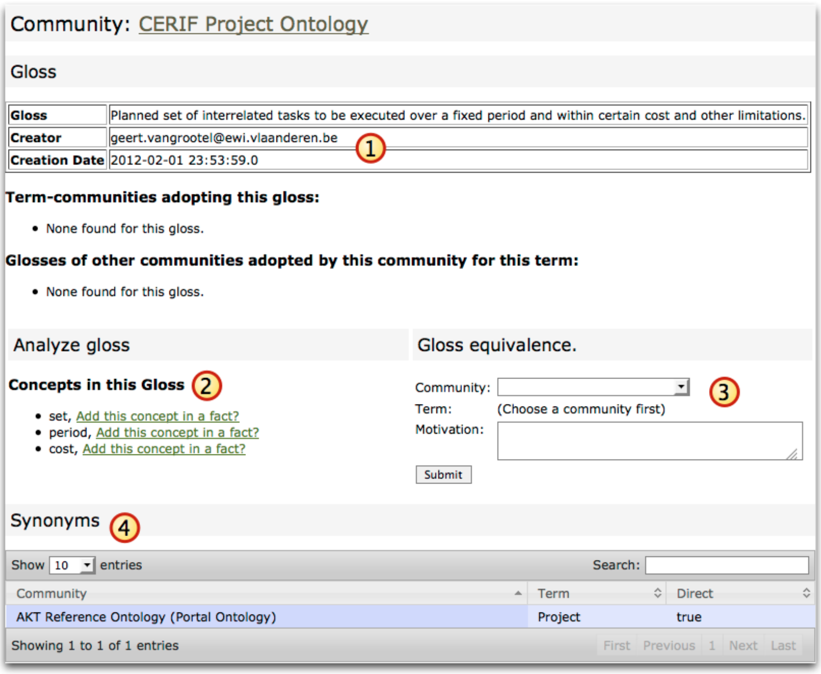

Fig. 5. Displaying the gloss of a community-term pair.

6 Discussion and Future Work

Every method needs to be teachable, repeatable and traceable. The GOSPL method for hybrid ontology engineering complies with all three criteria. The first two criteria have already been proven in industry; we went beyond the current state of affairs with the third criterion by formalizing the social processes involved. This allows us to store the whole dialogue within the community, supporting decision-making that could result in ontology evolution.

Teachable. The DOGMA framework for ontology engineering, on which GOSPL is based upon, is drew inspiration from database design methods and techniques such as NIAM and ORM. NIAM/ORM and therefore also DOGMA are fact-oriented approaches in which stakeholders communicate fact-types expressed in natural language. Fact-oriented approaches differ from frame-oriented approaches (e.g., UML) by eliminating the distinction between attributes and relations; every thing is a fact between concepts. This reduces the learning curve. Unlike UML, fact-orientation was not intended to capture the dynamic aspects of a system (e.g., methods). The use of natural language to express these fact-types also facilitates the knowledge elicitation processes.

Repeatable. GOSPL extends the ontology evolution processes of collaborative ontology engineering approaches DOGMA Meaning Evolution Support System (MESS) [5] and Business Semantics Management [2]. Both methods clearly defined the different processes in an iteration to extend the ontology. The repeatability of these methods has already been proved in the industry [2].

Traceable. In order to support ontology evolution, one needs to record the changes over time. As in software engineering, it is a good practice to also document why certain changes have been made. The different evolution operators on the formal parts are therefore traceable (who, why, when, etc.), what is not often captured is the whole process of reaching a decision, with GOSPL, the social processes leading to a change in the ontology will have been formalized and stored for future reasoning.

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. For a closer integration of other means of interaction such as teleconferences, we could draw inspiration from [16] where they presented a customizable collaborative environment focused to support ontology-based enterprise interoperability.

Following the results of [4] in which we analyzed the interactions between users and the system and were able to clustering types of users, we are currently investigating the use of data mining techniques to identify types of users. This will allow us to assign different workflows to the different types of users. Since GOSPL requires community-agreement before an ontology evolution operator is applied, one of the hypothesis we make is that terms, fact-types and its informal descriptions and constraints will undergo noticeably less changes over time.

7 Conclusion

The problem in ontology engineering is not on what ontologies are, but how they become operationally relevant and sustainable over longer periods of time, and how proper method and tool support can be provided. GOSPL extends the fact-oriented and layered ontology framework DOGMA and provides is a collaborative ontology engineering method that supports stakeholders in iteratively interpreting and modeling their ontologies in their own terminology and context. GOSPL formalizes the social processes that result in ontology evolution operators and uses a special linguistic resource containing definitions of concepts referred to by the community for high level reasoning amongst the human stakeholders. In this paper, we presented details on the GOSPL method and prototype for fact-oriented ontology engineering in which the ontology evolves only if the community agrees on the changes in a discussion. The fact-oriented models are automatically implemented into OWL/RDF(S) for use with off-the-shelf semantic technologies and the annotated data sources are exploited to guide the discussion.

Acknowledgements

This work was partially funded by the Brussels Institute for Research and Innovation through the Open Semantic Cloud for Brussels Project. The authors would like to thank their colleague Cristian Vasquez for his comments.

References

1. Bergman, M.: A brief survey of ontology development methodologies. Via http://www.mkbergman.com/906/a-brief-survey-of-ontology-development-methodologies/ (2010)
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. De Leenheer, P., Debruyne, C.: DOGMA-MESS: A tool for factoriented collaborative ontology evolution. In: On the Move to Meaningful Internet Systems 2008: ORM (ORM 2008). LNCS, Monterrey, Mexico, Springer (2008)
4. De Leenheer, P., Debruyne, C., Peeters, J.: Towards social performance indicators for community-based ontology evolution. In: Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2008), collocated with the 8th International Semantic Web Conference (ISWC 2009), CEUR-WS (2009)
5. de Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: Proc. of the 14th Int. Conf. on Conceptual Structures (ICCS 2006). Vol. 4068 of LNCS., Springer (2006) pp. 189–203
6. De Nicola, A., Missikoff, M., Navigli, R.: A software engineering approach to ontology building. Information Systems **34** (2009) pp. 258–275
7. Debruyne, C., Meersman, R.: Semantic interoperation of information systems by evolving ontologies through formalized social processes. In: ADBIS: Eder, J., Bieliková, M., Tjoa, A.M. (eds.). Vol. 6909 of LNCS., Springer (2011) pp. 444–459

8. Fernández-López, M., Gomez-Perez, A., Juristo, N.: METHONTOLOGY: from ontological art towards ontological engineering. In: Proceedings of the AAAI97 Spring Symposium. (1997) pp. 33–40
9. 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.
10. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies **43** (1993) pp. 907–928
11. Guha, R., Lenat, D.: Cyc: A midterm report. AI Magazine **11**(3) (1990) pp. 32–59
12. Halpin, T.: Information Modeling and Relational Databases. Morgan Kaufmann, San Francisco, CA, USA (2008)
13. Holsapple, C.W., Joshi, K.D.: A collaborative approach to ontology design. Commun. ACM **45** (2002) pp. 42–47
14. Karapiperis, S., Apostolou, D.: Consensus building in collaborative ontology engineering processes. Knowledge Creation Diffusion Utilization **1**(3) (2006) pp. 199–216
15. Kotis, K., Vouros, A.: Human-centered ontology engineering: The hcome methodology. Knowledge Information Systems **10**(1) (2006) pp. 109–131
16. Liapis, A., Christiaens, S., De Leenheer, P.: Collaboration across the enterprise - an ontology-based approach for enterprise interoperability. In: ICEIS (4): Cordeiro, J., Filipe, J. (eds.). (2008) pp. 255–262
17. 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
18. 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
19. Siorpaes, K., Simperl, E.: Human intelligence in the process of semantic content creation. World Wide Web **13**(1-2) (2010) pp. 33–59
20. 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
21. Uschold, M.: Building ontologies: towards a unified methodology. In: 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems. (1996) pp. 16–18
22. Uschold, M., Grüninger, M.: Ontologies: principles, methods, and applications. Knowledge Engineering Review **11**(2) (1996) pp. 93–155
23. 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
24. Vrandecic, D., Pinto, H.S., Sure, Y., Tempich, C.: The DILIGENT knowledge processes. Journal of Knowledge Management **9**(5) (2005) pp. 85–96
25. Wintraecken, J.: The NIAM Information Analysis Method, Theory and Practice. Kluwer Academic Publishers (1990)