Mind the Gap!
Transcending the Tunnel View on Ontology Engineering

Pieter De Leenheer and Stijn Christiaens
Semantics Technology & Application Research Laboratory (STARLab)
Vrije Universiteit Brussel - Pleinlaan 2 - B1050 BRUSSEL 5
{pieter.de.leenheer,stijn.christiaens}@vub.ac.be

ABSTRACT
The key objective of communal knowledge sharing at the scale of the World Wide Web is the ability to collaborate and integrate within and between communities. Ontologies, being formal, computer-based specifications of shared conceptualisations of the worlds under discussion, are instrumental in this process by providing shared semantic resources. To this end, the pragmatic aspects of the exchange of knowledge and information are crucial. Pragmatics represent the intentions, motivations and methodologies of the persons involved and need to become formalised and unambiguous for effective exchange to occur. On the one hand, this is something that humans manage fluently in their daily face-to-face social discourses. On the other hand, as contemporary knowledge engineering methods consider only the non-human system parts, they usually focus on mere syntactic aspects of concept modelling. The elicitation (semantics) and application (pragmatics) context are often weak or even ignored. This paper aims to bridge this gap between "reality" and its modelling concepts by (i) transcending knowledge engineering methods to a semiotics view on contextualised communal knowledge engineering and sharing; and (ii) by presenting the DOGMA ontology framework and how it provides extension points to this semiotics engineering.

Categories and Subject Descriptors
I.2.4 [**Knowledge Representation Formalisms and Methods**]: *representation languages, semantic networks*

General Terms
Management, Experimentation, Human Factors, Languages, Theory

Keywords
ontology, knowledge sharing, DOGMA, semiotics, context dependency management

1. INTRODUCTION

The key objective of communal knowledge sharing at the scale of the World Wide Web is the ability to collaborate and integrate within and between different and diverse communities. A community constitutes a social system, where action and discourse is performed within more or less well-established goals, norms, and behaviour [10]. Communication is the primary basis for coordinated action, hence in order to collaborate and integrate between different and diverse communities, it is important to capture and agree on the semantics of the concepts being communicated.

In order to fulfil the rapidly evolving community requirements, the concepts being communicated must continuously be adapted to the actual collaborative situation, and the meaning of new concepts should be incrementally negotiated by all participating stakeholders. Therefore, knowledge creation becomes the ultimate red-thread action transcending communities to develop consensus for communication in order to accomplish their goals.

Ontologies, being formal, computer-based specifications of shared conceptualisations of the worlds under discussion, are instrumental in this process by providing shared resources of syntax and semantics [7]. Such formal semantics are evidently fundamental in the development of any collaborative, knowledge-intensive services, methodologies or systems that claim to capture and evolve, in real time, relevant commonalities and differences in the way communities conceptualise their world and communicate about it.

To this end, the pragmatic aspects of the exchange of knowledge and information are crucial. Pragmatics represent the intentions, motivations and methodologies of the persons involved and need to become formalised and unambiguous for effective exchange to occur. This is something that humans manage fluently in their daily face-to-face discourses. However, in current practice, knowledge engineering methods usually focus on mere syntactic aspects of concept modelling, and the context of elicitation (semantics) and application (pragmatics) is often weak or even completely ignored. Furthermore, systems are usually reduced to only the non-human parts, with the possible exception of the field of organisational semiotics [16,25] that already involved a few socio-technical aspects of communities such as norms and behaviour in information system specification. Semiotics [4,9,19] is a science of signs and their syntax, semantics and pragmatics that, by giving an interdisciplinary, socio-technical view on an information system specification, aims to bridge this gap between "reality" and its modelling concepts [10].

Similarly, this paper wants to bridge the gap in a knowledge-intensive system between its social/human part (where knowledge is socialised in daily face-to-face discourses), and its technical part (where knowledge is engineered as pure mathematical objects) (Sect. 2). Doing so, it contributes to state-of-the-art in knowledge engineering methods by (i) transcending these methods to a semiotics view on system engineering for communal knowledge sharing by introducing requirements for semiotics engineering (Sect. 3); and (ii) by presenting the DOGMA ontology framework and how it provides extension points to this semiotics engineering (Sect. 4). Section 5 gives a real-world case study excerpt from engineering semiotics for competency modelling. Finally, we end with a discussion and conclusion.

2. COMMUNITIES AND KNOWLEDGE

In order to illustrate the gap we mentioned between the social/human knowledge-sharing system and its technical "mirror", we adopt Nonaka's well-known four modes of knowledge conversion [18] (Fig. 1). Critics have argued that Nonaka and Takeuchi's distinction between tacit and explicit knowledge is oversimplified and that the notion of explicit knowledge is self-contradictory [12]. Specifically, for knowledge to be made explicit, it must be translated into information (i.e., symbols outside of our heads) and knowledge, which is what semiotics engineering is all about.

Communication is the primary basis for coordinated action for accomplishing community goals. Through **socialisation,** people naturally *utter* and *share* experience and expertise in face-to-face discourses, and thereby create *tacit* knowledge such as mental models of ontological concepts and new technical skills. The current application context is about the concept Deliver.

Through **externalisation**, this tacit knowledge is *partly* articulated (publicly or privately) into explicit formal knowledge *artefacts*, taking the shape of e.g., a concept type, a contributed taxonomy, an interface, a workflow definition, etc. This is illustrated by the curved arrows that take a selection from the mental models that is relevant to explicate the concept Deliver in this application context. Note that although externalisation is an incremental process using language of variable expressivity (as we will explain below), there will inevitably remain an important part of tacit knowledge in the utterer's mental model on which the correct interpretation of the articulated part is dependent [20,21]. Externalisation is done by domain experts, as they have the tacit knowledge about the domain and can sufficiently assess the real impact of the conceptualisations and derived collaborative services on their organization.

Once they are published, **combination** involves semantic analysis and integration (for an excellent survey, see [15]) of published contextualised knowledge artefacts in order to adapt to new collaborative requirements. This process might be further constrained by community-shared models such as running application tools that commit to certain published consensus, by pre-existing organisational sub-ontologies, and by inflexible data schemas interfacing to legacy data. Furthermore, participating stakeholders usually have strong individual interests, inherent business rules, and entrenched work practices that influence decisions in meaning negotiation rounds. These may be tacit, or externalised in workflows that are strongly interdependent, hence further complicate the conceptual combination. Sometimes it is not necessary (or even possible) to reach for context-independent ontological knowledge, as most ontologies used in practice assume a certain context and perspective of some community [23]. Wenger [26] supported this by stating *"Peace, happiness, and harmony are therefore not necessary properties of a community of practice"*. Hence pragmatically, combination processes need to support human experts to focus on these "community-grounded" processes of realising the appropriate amount of consensus on *relevant* conceptual definitions through *effective* meaning negotiation in an *efficient* manner [7].

Internalisation concerns the appropriate operationalisation and embodiment of explicit knowledge consensus in the current communication actions. For example, for ontological knowledge, the most widely used recommendations on the Semantic Web are XML, RDF(S) and OWL.

The four modes of knowledge conversion engender an upward knowledge spiral, where individual knowledge opinions become commonly accepted, through an iterative interplay between externalisation and internalisation. This interplay illustrates how knowledge artefacts co-evolve with their communities of use. In socialisation, knowledge is communicated as abstract entities, and humans can easily grasp the context of interpretation. However, when applying formal knowledge engineering methods for the other knowledge conversion modes, we have to emphasize the semiotic dimensions, in order to bridge the gap between socialisation processes in the social system on the one hand, and the other conversion processes (externalisation, internalisation, and combination) in the technical system on the other hand.

In the next section, we bootstrap some requirements for semiotics engineering that address the context of elicitation (semantics) and application (pragmatics).

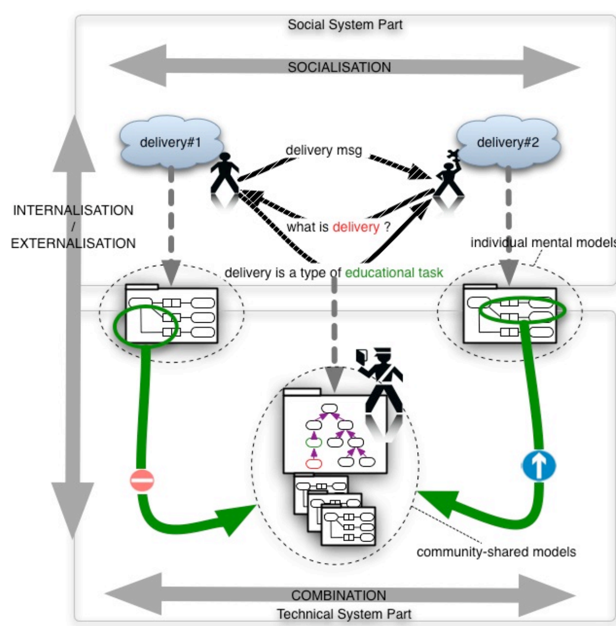

Figure 1: The gap between the social/human knowledge-sharing system and its technical "mirror".

3. REQUIREMENTS FOR SEMIOTICS ENGINEERING

Semiotics engineering (as introduced by Zhao [27]) concerns the process of engineering a symbolic system that formalises models of data, processes, and ontological knowledge in well-formed symbolics. Similarly to ontology engineering, we distinguish two directions of semiotics engineering: *elicitation* and *application* [6].

Knowledge elicitation is related to Nonaka's processes of externalisation and combination. Guidance for elicitation is triggered and influenced by composition norms that hold in the community [8]. Elicitation concerns the syntactic and semantic dimensions of semiotics. Knowledge application is related to combination and internalisation, and concerns the pragmatic dimensions of semiotics.

From a pragmatic point of view, *scalable* elicitation should on the one hand be driven by the intended purpose or application, but should on the other hand result in reusable knowledge artefacts. In [6], we identified a set of context-driven ontology engineering processes that govern this trade-off. Considering the literature, these processes differ widely in their implementation. The only constraint we stress here is that their user interfaces should be accommodated for ease of use in communities.

3.1 Constructivism

Constructivism rejects the existence of a unique objective reality, hence its reflecting "transcendent" conceptualisation. Therefore, the *constructivist* approach supports multiple domain experts in the gradual and continuous externalisation of their subjective realities contingent on their ever-evolving social and cultural background, and professional experience. Technically this requires a knowledge engineering methodology that supports the building and managing of increasingly mature versions of ontological artefacts (conceptualising their divergent subject realities), and of their converging interrelationships, achieved through careful negotiation.

Wittgenstein and Putnam also consider the meaning of a concept to be the set of all its uses (read: application contexts) [22, pp.128]. Pragmatics boils down to converging to that subject conceptualisation that maximally fits the intended application context.

3.2 Variability and Reusability

All meaning (semantics) is for communication purposes about a universe of discourse. It is represented independent of language but necessarily must be entirely rooted and described in (natural) language. Linguistic "grounding" of meaning is achieved through elicitation contexts, which can be mappings from identifiers to source documents such as generalised glosses, often in natural language [6,14].

Thousands of shared vocabularies or so-called *folksonomies* emerge, are sold and advertised, prosper or wither in a self-organising manner on Web 2.0, through reuse and adaptation of natural language labels for tagging their resources, such as process and work flow models. Natural language labels for concepts and relationships bring along their inherent *ambiguity* and *variability* in interpretation [2], which on the one hand provides an *unbounded reusability* potential for specific reference in a given application context, which is important for *scalable* semiotics engineering [27]. On the other hand, however, it requires an analysis of multiple contexts to conduct successful *lexical disambiguation* on the labels [1,5].

3.3 Application-specific Contextualisation

Tagging resources and thereby externalizing a cloud of lexically disambiguated concept labels is not enough. From a pragmatic point of view, elicitation must anticipate on the intended application by eliciting and combining the relevant artefacts insofar necessary in an effective way. In Section 3.2, we stressed the unbounded potential for vocabulary reusability and scalability, however true knowledge creation for the benefit of the community involves further externalisation of specific concept semantics such as attributes and axioms, uniformly agreed, but properly contextualised for a particular application context.

3.4 Flexible Operationalisation

Once (a version of) an artefact or ontology has been agreed on and validated, it can be translated into an operational language that is in accordance with the actual collaborative application pool and internalised in the tacit mental models. For example, the most widely used recommendations on the Semantic Web are XML, RDF(S) and OWL. However, as community goals tend to shift depending on the changing shared business interests, an operationalised ontology version (or even its operational language) will soon become obsolete. An ontology (language) should capture these changes continuously in order to co-evolve driven by the ontology engineering activities described so far.

To formalise the semiotics engineering framework we circumscribed above, we adopt and extend the DOGMA[1] ontology engineering approach.

4. DOGMA ONTOLOGY ENGINEERING

Ontology is an approximate shared semiotic representation of a subject matter. To fulfil the requirements mentioned in previous section, the DOGMA [6,13,17,24] ontology approach and framework is adopted with the intention to create flexible, reusable bounded semiotics for very diverse computational needs in communities for an unlimited range of pragmatic purposes [27].

The DOGMA approach has some distinguishing characteristics that make it different from traditional ontology approaches such as (i) its groundings in the linguistic representations of knowledge, (ii) the explicit separation of the *conceptualisation* (i.e., lexical representation of concepts and their inter-relationships, materialised by so-called *lexons*) from its *axiomatisation* (i.e., semantic constraints) and (iii) its independence from a particular representation language. The goal of this separation, referred to as the *double articulation* principle [24], is to enhance the potential for re-use and design scalability.

[1] acronym for Developing Ontology-Grounded Methods and Applications

Lexons are initially uninterpreted binary fact types, hence underspecified, which increases their potential for reusability across community perspectives or goals [2]. The axiomatisation of lexons guarantees the specification needed for semantic consistency and well-formedness in a particular collaborative context (see further).

Lexons are collected in the Lexon Base, a reusable pool of possible vocabularies. A lexon is a 5-tuple declaring either (in some *elicitation context* G) [6]:

1. taxonomical relationship (genus): e.g., < G, manager, is a, subsumes, person >;
2. non-taxonomical relationship (differentia): e.g., < G, manager, directing, directed by, company >.

Lexons could be approximately considered as a combination of an RDF/OWL triple and its inverse, or as a conceptual graph style relation [22].

Next, we will elaborate more on the notions of elicitation context (Sect. 4.1) and application context (Sect. 4.2).

4.1 Language versus Conceptual Level

Another distinguishing characteristic of DOGMA is the explicit *duality* (orthogonal to double articulation) in interpretation between the language level and conceptual level. The goal of this separation is primarily to disambiguate the lexical representation of terms in a lexon (on the language level) into concept definitions (on the conceptual level), which are word senses taken from lexical resources such as WordNet [11]. The meaning of the terms in a lexon is dependent on the context of elicitation [5].

For example, consider a term “capital”. If this term was elicited from a typewriter manual (read: context *G*), it has a different meaning (read: concept definition) than when elicited from a book on marketing. The intuition that a context provides here is: a context is an abstract identifier that refers to implicit and tacit assumptions in a Universe of Discourse (UoD), and that maps a term to its intended meaning (i.e. concept identifier) within these assumptions [6]. Notice that a context in our approach is not explicit formal knowledge. In practice, we externalise an elicitation context by referring to a source (e.g., a set of documents, laws and regulations, informal description of best practice, etc.), which, by *human understanding*, is assumed to “contain” the necessary assumptions [13].

Hence, within a context of elicitation, a lexon is not merely syntactic by nature. They are just underspecified, what makes them reusable for being applied in specific collaborative application contexts within a UoD. The formal account for *application context* is manifested through the selection and interpretation of lexons in *ontological commitments*, and the *context dependencies* between them [6].

4.2 Ontological Commitments

The pragmatic account for knowledge artefacts is formalised in ontological commitments. Committing to the Lexon Base in the *context of an application* means selecting a meaningful set *S* of lexons from the Lexon Base that approximates well the intended vocabulary, followed by the addition of a set of semantic constraints, or rules, to this subset. The result (i.e., *S* plus a set of constraints), called an ontological commitment, is a logical theory of which the models are first-order interpretations that correspond to the intended task(s) for achieving a particular goal with a certain level of trust and quality. An important difference with the underlying Lexon Base is that commitments are internally unambiguous and semantically consistent[2]. Though ontologies can differ in syntax, semantics, and pragmatics, they all are built on this shared vocabulary, called the Lexon Base. Examples of ontological commitments are business rules, database constraints, or norms.

4.3 Context Dependency Management

The motivation to regard context dependencies as a facet of an elicitation or application context comes from classical literature, where context is adopted for packaging, disambiguating, linking, and nesting knowledge artefacts [6]. Doing so, two formal artefacts might be mutually inconsistent, and at the same be dependent on the same upper core artefact. When the latter changes, the dependencies will force the dependents to evolve along. Context dependencies provide additional information on the pragmatic context of the dependent.

5. COMPETENCY MODEL BUILDING

In this section we demonstrate how the requirements for semiotics engineering are facilitated by our DOGMA approach, with an example from the Human Resource domain.

5.1 Reusable Competence Definitions

The Lexon Base provides possible vocabularies that can be reused as building blocks for integration. We consider three stakeholding communities: the Superb Actors School (SAS), the Public Employment Agency (PEA), and the RCD Vocabulary Advisory (RVA). All of them commit to the Lexon Base, of which a sample is given in Table 1. This means that their ontological commitments use a vocabulary in which all terms are linguistically grounded and lexically disambiguated[3]. RVA is

---

[2] Although it is outside the scope of this article, we find it valuable to note that in the research community it is debated that consistency is not necessarily a requirement for an ontology to be useful.

[3] E.g., the term “action” is disambiguated as in [10]:

An action is a transition involving a non-empty set of actors in its pre-state, and, if not "destroyed" or "consumed" by the action, in its post-state as well, and involving a nonempty or empty set of other things (actands) as part of its pre-state, and having a nonempty or empty set of other things (actands) in its post-state. Examples:

1. Stock-taking (action) by a warehouse-clerk (actor) checking current stock, and producing a stockinventory;
2. Issuing a stock item (action) by a stock supervisor (actor), resulting in a change of stock level;
3. Writing (action) a report by an author (actor);

responsible for maintaining reusable vocabularies for competence definitions. Note that some lexons such as "speech clarity is_a /subsumes competence" imply a taxonomy.

Table 1: a sample Lexon Base

G[4]	Head term	Role	Co-role	Tail term
RVA_vocabulary_20060410	Competence	consists_of	part_of	Action
RVA_vocabulary_20060410	Competence	belongs_to	has_competence	Actor
…	…	…	…	…
Actor_School_20070514	Speech_Clarity	has_quality	quality_of	Very_good
Actor_School_20070514	Speech_Clarity	consists_of	part_of	Speak
Actor_School_20070514	Speech_Clarity	results_in	result_of	Understand
Actor_School_20070514	Speech_Clarity	is_a	subsumes	Competence
…	…	…	…	…
Public_Employment_20070514	Person_Actor	is_a	subsumes	Actor
Public_Employment_20070514	Speech_Clarity	belongs_to	has_competence	Person_Actor
Public_Employment_20070514	Speech_Clarity	belongs_to	has_competence	Politician
…	…	…	…	…

5.2 Contextualised Competency Building

An important collaborative goal of both communities is to enhance job matching for actors. Knowledge workers, including domain experts and (middle) managers are responsible for detecting and reporting flaws in the job matching processes caused by a lack of consensus about the structure, meaning and use of concepts. The knowledge engineer interprets then these reports and formulates them in concrete externalisation tasks, which are after initiation delegated to the relevant participants for execution. To this end, communities can externalise their knowledge and combine it with RVA vocabulary insofar necessary and relevant for the communication context at stake.

For example, consider the elicitation of a competency model for a key competence "Speech_Clarity" in job matching. Before this concept is elicited from scratch or by making uncontrolled subjective selections from the Lexon Base, Ontology Server is queried for existing ontological commitments that already formalise the concepts "Competence" and/or "Speech_Clarity" for some application context both stakeholding communities share.

In general, the Ontology Server hosts the different knowledge repertoires of collaborating communities. Each repertoire is contextualised: it stores different kinds of knowledge artefacts, and sometimes multiple structures or semantics for the same artefact can exist.

In this case, if the artefacts already exist, they can be reused as the context for the current elicitation tasks in this new application context between SAS and PEA. However, it turns out that:

1. "Speech_Clarity" was not yet formalised so far, but for "Competence", it appears that SAS and PEA share a minimal but unambiguous commitment, consisting of one axiomatised lexon formalising following business rule *BR*:

```
Competence belongs_to AT LEAST ONE Actor.
```

2. "Speech_Clarity" is a sub-type of "Competence", hence the (diverging) educational and employment commitments for "Speech_Clarity" from SAS and PEA respectively, will be context-dependent on this shared commitment.

SAS' speech clarity The concepts in the SAS community are more detailed when it comes to competence descriptions. Since they are responsible for making sure their students obtain the competences, they need a clear view on what the competences *mean*. Figure 2 shows their ontological commitment (to the Lexon Base) on "Speech_Clarity".

PEA's speech clarity The PEA community is focused on matching competences with jobs. For them, it is less relevant to have a detailed view on the competences themselves, but rather on the link between competences and possible job types. Figure 3 displays their commitment on "Speech_Clarity".

It is clear that SAS and PEA have a different use of the concept "Speech_Clarity", though both are based on the same business rule *BR*. However, the focus in this paper was not to force both stakeholders to merge to one objective concept of "Speech_Clarity", but to eamphasise the importance of dependencies that exist between different contexts of use of the same or taxonomically related concepts "Competence" and "Speech_Clarity".

5.3 Context Dependencies

As already mentioned both PEA's and SAS' commitments are context-dependent on the shared business rule *BR*. Suppose the administrator of *BR* decides to change the lexical representation or alter the semantics of the concepts (read: lexons) involved, then this change must be propagated to all its dependent contexts of use.

Furthermore, as part of its ontological commitment, the PEA also creates a context dependency between its ontology and the ontology from the SAS. This dependency states that the PEA

4. Expressing (action) a conception by a person (actor), in the form of a representation.

[4] The elicitation context identifier G in our example is derived from space-time dimensions: (1) the community that elicited the lexon, and (2) the date on which this version was created.

system should be updated whenever there are changes related to the concepts "Person_Actor" and "Speech_Clarity" in the SAS system. When the SAS ontology administrators update their world view (e.g., by changing "Speech_Clarity" to "On_Stage_Speech_Clarity"), the technical system *triggers* the social system between the two communities. Based on norms the appropriate knowledge workers in the PEA community receive a request for a knowledge change, stating that they should check and update their ontology (e.g. by taking over "On_Stage_Speech_Clarity" which they link to "Person_Actor", but not to "Operator").

In this example, we showed an application of some of the processes depicted in Fig. 1. We demonstrated combination by merging a part of the SAS context with the PEA context ("Speech_Clarity" belongs_to "Person_Actor"). In the social part, the *socialisation* was demonstrated through the evolutionary trigger stated by the context dependency, where the PEA administrators had to understand and reflect the change in the SAS context. This change also showed *internalisation* as the administrators analysed this external knowledge, studied and understood the change. They *externalised* this by formalising the change in their own context.

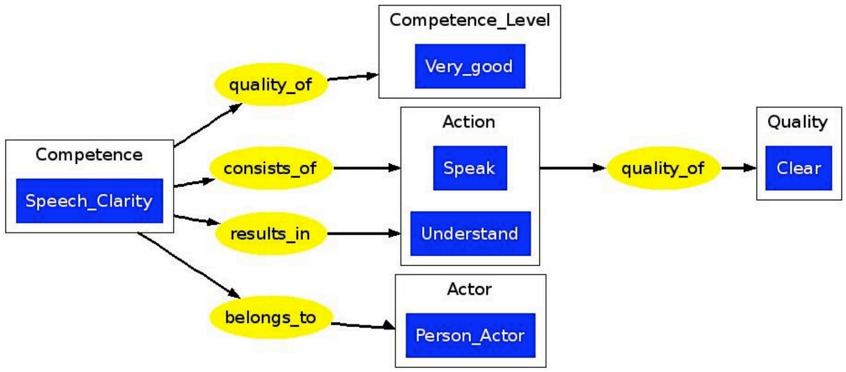

Figure 2: SAS' educational commitment

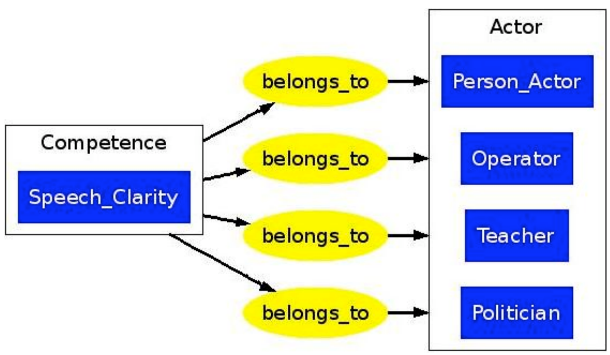

Figure 3: PEA's employment commitment

6. IMPLEMENTATION

DOGMA[5] Studio is the tool suite behind the DOGMA ontology engineering approach. It contains both a Workbench and a Server. The Workbench is constructed according to the plug-in architecture in Eclipse. There, plug-ins, being loosely coupled ontology viewing, querying or editing modules support the different semiotics engineering activities and new plug-ins continuously emerge. This loose coupling allows any arbitrary knowledge engineering community to support its own ontology engineering method in DOGMA Studio by combining these plug-ins arbitrarily. Such a meaningful combination of view/edit/query plug-ins is called a "perspective" in Eclipse. The DOGMA Server is an advanced J2EE application running in a JBoss server, which efficiently stores Lexons and Commitments in a PostgreSQL Database. DOGMA Studio is complemented by a community layer, in which the DOGMA collaborative ontology engineering processes are grounded in communities of use. This layer is implemented by the DOGMA-MESS[6] methodology and system. For an in-depth elaboration on DOGMA Studio and –MESS in the context of a business use case, we refer to [3].

7. DISCUSSION AND CONCLUSION

Successful virtual communities and communities of stakeholders are usually self-organising knowledge-intensive systems. The knowledge creation and sharing process is driven by implicit community goals such as mutual concerns and interests. Current knowledge engineering methods usually focus merely on the syntactic dimension of semiotics engineering, thereby ignoring these pragmatic aspects. In order to better capture relevant knowledge in a community-goal-driven way, these community goals must be externalised appropriately. In this paper we considered a knowledge-intensive system as a true semiotic system, consisting of a social/human part and a non-human technical part. We proposed an initial, non-exhaustive list of requirements for semiotics engineering where:

[5] http://starlab.vub.ac.be/website/dogmastudio

[6] http://www.dogma-mess.org

1. underspecified vocabularies promote unbounded reusability potential for specific ontological reference in a given application context. In our example, the linguistic grounding and disambiguation of these vocabularies are authored by a designated stakeholder in the community;
2. stakeholders agree on and commit to concepts insofar necessary and relevant for their communication, hence knowledge socialisation in a particular application context;
3. context dependencies between knowledge artefacts provide additional context information between these knowledge artefacts. If a knowledge artefact changes, a change request must be triggered to the authors of all knowledge artefacts that are dependent on it.

The current community goals and norms may be linked to relevant strategies underlying the legitimate collaborative knowledge conversion processes and its support. This requires us to model communities completely (i.e. establish their formal semantics) in terms of their intrinsic aspects such as goals, actors, roles, strategies, workflows, norms, and behaviour, and to so integrate the concept of community as first-class citizen in the knowledge structures of the evolving system. This holistic approach is breaking with current practice, where systems are usually reduced to only the non-human parts, with the possible exception of the field of organisational semiotics that already involved a few socio-technical aspects of communities such as norms and behaviour in (legitimate) information system specification.

8. ACKNOWLEDGMENTS

We would like to thank Robert Meersman and Aldo de Moor for the valuable discussions about theory and case. This research has been partially funded by the EU Leonardo da Vinci Project CODRIVE (BE/04/B/F/PP-144.339) and the Brussels Capital Region (IWOIB – PRB 2006).

9. REFERENCES

[1] Bachimont, B., Troncy, R., Isaac, A. (2002) Semantic commitment for designing ontologies: a proposal. In Gómez-Pérez, A., Richard Benjamins, V., eds.: Proc. of the 13th Int'l Conf. on Knowledge Engineering and Knowledge Management. Ontologies and the SemanticWeb (EKAW 2002) (Siguenza, Spain), Springer Verlag, pp. 114–121

[2] Bouaud, J., Bachimont, B., Charlet, J., and Zweigenbaum, P. (1995) Methodological Principles for Structuring an "Ontology". In Proc. IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing" (Montreal, Canada)

[3] Christiaens, S., De Leenheer, P., de Moor, A., Meersman, R. (2007) Business Use Case: Ontologising Competencies in an Interorganisational Setting. In Ontology Management: Semantic Web, Semantic Web Services, and Business Applications, from Semantic Web and Beyond: Computing for Human Experience, eds. Hepp, M.; De Leenheer, P.; de Moor, A.; Sure, Y., Springer

[4] Culler, J. (1981) The Pursuit of Signs: Semiotics, Literature, Deconstruction. Cornell University Press

[5] De Leenheer, P., de Moor, A. (2005) Context-driven disambiguation in ontology elicitation. In Shvaiko, P., Euzenat, J., eds.: Context and Ontologies: Theory, Practice, and Applications. Proc. 1st Context and Ontologies Workshop, AAAI/IAAI 2005 (Pittsburgh, USA), pp. 17–24

[6] De Leenheer, P., de Moor, A., and Meersman, R. (2007) Context Dependency Management in Ontology Engineering: a Formal Approach. Journal on Data Semantics VIII, LNCS 4380, Springer-Verlag, pp. 26-56

[7] de Moor, A., De Leenheer, P., and Meersman, R. (2006) DOGMA-MESS: A meaning evolution support system for interorganisational ontology engineering. In Proc. 14th Int'l Conf. Conceptual Structures (ICCS 2006) (Aalborg, Denmark), LNAI 4068, Springer Verlag, pp 189–203

[8] de Moor, A. and Weigand, H. (2007) Formalizing the evolution of virtual communities. In Information Systems 32(2):223-247

[9] De Saussure, F. (1966) Course in General Linguistics. McGraw-Hill

[10] Falkenberg, E.D.; Hesse, W.; Lindgreen, P.; Nilsson, B.E.; Oei, J.L.H.; Rolland, C.; Stamper, R.K.; Van Assche, F.J.M.; Verrijn-Stuart, A.A.; Voss, K. (1996) FRISCO : A Framework of Information System Concepts, The IFIP WG 8.1 Task Group FRISCO

[11] Fellbaum, C., ed. (1998) Wordnet, an Electronic Lexical Database. MIT Press

[12] Gourley, S., and Nurse, A. (2005) Flaws in the Engine of Knowledge Creation: a Critique of Nonaka's Theory. In Challenges and Issues in Knowledge Management, Information Age Publishing, pp. 293-315

[13] Jarrar, M., Demey, J., Meersman, R. (2003) On reusing conceptual data modeling for ontology engineering. Journal on Data Semantics 1(1):185–207

[14] Jarrar, M. (2006) Position paper: towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering. In Proc WWW 2006. ACM Press, pp. 497-503

[15] Kalfoglou, Y., Schorlemmer, M. (2005) Ontology mapping: The state of the art. In Proc. Dagstuhl Seminar on Semantic Interoperability and Integration (Dagstuhl, Germany)

[16] Liu, K. (2000). Semiotics in information systems engineering. Cambridge, England: Cambridge University Press

[17] Meersman, R. (1999) Semantic Ontology Tools in IS Design. In Proc. of the International Symposium on Methodologies for Intelligent Systems (ISMIS 1999), pp. 30-45.

[18] Nonaka, I. and Takeuchi,, H. (1995) The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation. Oxford University Press

[19] Peirce, C.S. (1955) Logic as Semiotic: the Theory of Signs. In Bucher, J. (ed.) Philosophical Writings of Peirce. New York

[20] Polanyi, M. (1966) The tacit dimension. Routledge & Kegan Paul, London, UK

[21] Polanyi, M. (1969) Knowing and being. Routledge & Kegan Paul, London, UK

[22] Sowa, JF (1984). Conceptual structures: information processing in mind and machine. Addison-Wesley, Reading, Massachusetts

[23] Schoop, M.; de Moor, A.; Dietz, J. (2006) The pragmatic web: a manifesto. In Communications of the ACM 49(5):75-76

[24] Spyns, P.; Meersman, R.; Jarrar, M. (2002). Data Modeling versus Ontology Engineering. SIGMOD Record, 31(4), pp. 12–17

[25] Stamper, R. K. (2001). Organisational semiotics: Informatics without the computer? In K. Liu, R. J. Clarke, P. Bøgh Andersen, & R. K. Stamper (Eds.), Information, organisation and technology: Studies in organisational semiotics (pp. 115-171). Boston, MA: Kluwer Academic Publishers

[26] Wenger, E. (1998) Communities of practice: Learning, meaning and identity. Cambridge University Press, Cambridge, UK.

[27] Zhao, G. (2004) Application Semiotics Engineering. In Proc. of the 16th Int'l Conf. on Software Engineering & Knowledge Engineering (SEKE 2004) (Banff, Alberta, Canada), pp. 354-359