Towards Community-based Evolution of Knowledge-intensive Systems ⋆

Pieter De Leenheer and Robert Meersman

Semantics Technology and Applications Research Laboratory (STARLab)
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 BRUSSELS 5, Belgium
{pdeleenh,meersman}@vub.ac.be

Abstract. This article wants to address the need for a research effort and framework that studies and embraces the novel, difficult but crucial issues of adaptation of knowledge resources to their respective user communities, and *vice versa*, as a fundamental property within knowledge-intensive internet systems. Through a deep understanding of *real-time*, community-driven evolution of so-called ontologies, a knowledge-intensive system can be made operationally relevant and sustainable over longer periods of time. To bootstrap our framework, we adopt and extend the DOGMA ontology framework, and its community-grounded ontology engineering methodology DOGMA-MESS, with an ontology that models community concepts such as business rules, norms, policies, and goals as first-class citizens of the ontology evolution process. Doing so ontology evolution can be tailored to the needs of a particular community. Finally, we illustrate with an example from an actual real-world problem setting, viz. interorganisational exchange of HR-related knowledge.

1 Introduction

Collaboration and knowledge sharing have become crucial to enterprise success in the knowledge-intensive European Community and the globalised market world-wide. In this market the trend in innovation of products and services is shifting from mere production excellence to intensive and meaningful knowledge creation and management.

In next-generation computerised distributed working environments, a key objective indeed is to effectively leverage individual competencies of people working together to a *community level*. The World Wide Web has been extremely successful in enabling information sharing among a seemingly unlimited number of people worldwide. It therefore also provides the basic infrastructure that allows on-line virtual communities (professional as well as leisure-oriented) to emerge all around.

Currently, we are witnessing what some call "second-generation Web" (*Web 2.0*), manifested by an explosion of new tools and technologies being developed and shared

⋆ We would like to thank Stijn Christiaens, Aldo de Moor, Tom Mens, Stijn Hoppenbrouwers, and Erik Proper for the valuable discussions about the subject. The research described in this paper was partially sponsored by the EU Leonardo da Vinci CODRIVE project and the EU FP6 IST PROLIX project.

at little or no cost. Social applications like lightweight folksonomies, blogs, wikis, and a plethora of other collaborative tools yield value-added communication platforms that enable virtual communities to emerge that share ideas, knowledge and resources in a usually *self-organising* manner [33]. Even as we limit ourselves (as we do in this article) to professional, "goal-oriented" communities, the logical and inevitable next step is an increase in scale and maturity of such communal knowledge sharing, achieved through collaboration and integration within and between different and diverse communities.

Ontologies, being formal, computer-based specifications of *shared conceptualisations* of the worlds under discussion, are instrumental in this process by providing shared resources of *semantics* [18, 17, 22]. 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 article wants to address the need for a research effort and experimental framework that studies and embraces the novel, difficult but crucial issues of adaptation of knowledge resources to their respective user communities, and *vice versa*, as a fundamental property within knowledge-intensive internet systems. Through a deep understanding of *real-time*, community-driven evolution of so-called ontologies, a knowledge-intensive system can be made operationally relevant and sustainable over longer periods of time.

To bootstrap a framework, we adopt and extend the DOGMA ontology framework, and its community-grounded ontology engineering (OE) methodology DOGMA-MESS, with an ontology that models community concepts such as business rules, actors, roles, norms, and goals as first-class citizens of the ontology evolution process. Doing so ontology evolution can be tailored to the needs of a particular community. Finally, we illustrate with an example from an actual real-world problem setting, viz. interorganisational exchange of HR-related knowledge.

2 Progress beyond the State of the Art

Several EU FP6 integrated projects[1] and networks of excellence[2] tested and validated a vast number of methods and tools for formalising and applying knowledge representation models in a wide variety of applications [18, 17, 22]. However, there is still little understanding of, and technological support for, the methodological and *evolutionary* aspects of ontologies as resources. Yet these are crucial in distributed and collaborative settings such as the Semantic Web, where ontologies and their communities of use naturally and *mutually* co-evolve.

[1] e.g., http://www.sekt-project.com, http://dip.semanticweb.org
[2] e.g., http://knowledgeweb.semanticweb.org

2.1 Single User Ontology Evolution

For managing the evolution of domain vocabularies and axioms by one single dedicated user (or a small group under common authority), established techniques from data schema evolution [1, 19] have been successfully adopted, and consensus on a generic ontology evolution process model has begun to emerge [21, 25]. In Fig. 1, we illustrate a single user, hence *context-independent* change process model, based on [2], that distinguishes four activities, over three phases in the change process model: *initiation*, *execution*, and *evaluation*. For a comprehensive state-of-the-art survey on ontology evolution activities we refer to [7].

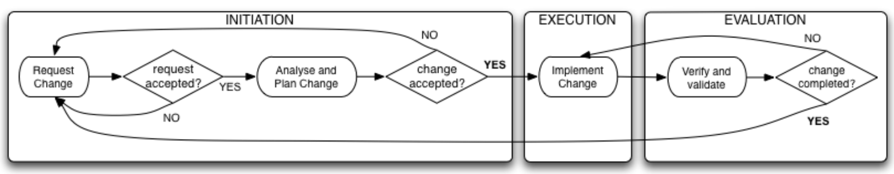

Fig. 1. A context-independent change process model.

Initiation *Requesting* the change has to do with initiating the change process. Some human stakeholder or automatic discovery process [30] wants to make a change to the ontology under consideration for some reason, and will post a so-called change request. Usually a change request is formalised by a finite sequence of elementary change *operations* [1]. The set of applicable change *operators* to conduct these change operations is determined by the applied knowledge representation model.

Planning the change has to do with understanding *why* and *where* the change needs to be made. Therefore, a crucial part of this activity has to do with *change impact analysis*, which is "the process of identifying the potential consequences (side effects) of a change, and estimating what needs to be modified to accomplish a change" [3]. This is very helpful to estimate the required cost and effort (see [27] for a business view on ontology engineering costs). A result of this activity may be to decide to implement the change, to defer the change request to a later time, or to ignore the change request altogether.

Execution The execution of a change request should have transactional properties, i.e., atomicity, consistency, isolation, and durability [16]. Our process model realises these requirements by strictly separating the change request specification and subsequent implementation, as suggested by [30]. Implementing a change is a difficult process that necessitates many different sub-activities: *change propagation*, *restructuring* and *inconsistency management*. Furthermore, different *evolution strategies* might be implemented to resolve inconsistencies during a change operation [1, 31, 23].

Evaluation The last, but certainly not the least, activity in the change process has to do with *verification* and *validation*. Verification addresses the question “did we build the product right?”, whereas validation addresses the question “did we build the right product?”. A wide scale of different techniques has been proposed to address these questions, including: testing, formal verification, debugging, and quality assurance.

2.2 Collaborative Ontology Engineering

Collaboration aims at the accomplishment of shared objectives and an extensive coordination of activities [26]. In order to create synergy in the result of the collaborative OE process, the *socio-technical* aspects of the community play a very important role [8]. E.g., through implicit and explicit norms, the authority for the control of the process is distributed among many different participants.

In a collaborative setting, there are many additional complexities that should be considered. As investigated in FP6 integrated projects[3] on collaborative networked organisations, the different professional experiences; social and cultural backgrounds among communities and organisations can lead to misconceptions, leading to frustrating and costly ambiguities and misunderstandings if not aligned properly. This is especially the case in interorganisational settings, where there may be many pre-existing organisational sub-ontologies, inflexible data schemas interfacing to legacy data, and ill-defined, rapidly evolving collaborative requirements [10]. Furthermore, participating stakeholders usually have strong individual interests, inherent business rules, and entrenched work practices. These may be tacit, or externalised in workflows that are strongly interdependent, hence further complicate the conceptual alignment.

Summarising, one should not merely focus on the practice of creating ontologies in a project-like context, but view it as a continuous process that is integrated in the operational processes of the community. In a collaborative setting, the *shared background* of communication partners is continuously negotiated as are the characteristics or values of the concepts that are agreed upon. The shared background is externalised as formal *artefacts*, which can be ontological elements of various levels of granularity, ranging from individual concepts of definitions to full ontologies, contributing to the communal knowledge. This includes contributed taxonomies, concept definitions, interfaces, workflow definitions, etc. The evolution of this shared background should be *orchestrated* by and *grounded* in the community.

2.3 Towards Community-based Evolution of Knowledge-intensive Systems

Successful virtual communities and communities of stakeholders are usually self-organising. The knowledge creation and sharing process is driven by implicit community *goals* such as mutual concerns and interests [24]. In order to better capture relevant knowledge in a *community-goal-driven* way, these community goals must be externalised appropriately. E.g., in [5], we identified several *macro-level* ontology engineering *processes* that (in a particular methodological combination) provide the goal of the ontology engineering process: *lexical grounding and disambiguation*, *specialisation*,

[3] e.g. http://ecolead.vtt.fi/

integration (including *negotiation*), *axiomatisation*, and *operationalisation*. However, in their operational implementation, which we call OE *micro-processes*, methodologies differ widely. In order to link the community goals to relevant *strategies* underlying the collaborative ontology engineering process and its support, we are required to model relevant community aspects (i.e. establish their formal semantics), and ultimately integrate the concept of community as first-class citizen, where possible, in the evolution processes of the knowledge-intensive system.

This *holistic* approach is breaking with current practice, where systems are usually reduced to only its IT aspects, with the possible exception of the field of *organisational semiotics* (e.g., MEASUR [29]) and the *language/action perspective* (e.g., RENESYS [8]) that already involved a few socio-technical aspects of communities such as norms and behaviour in legitimate user-driven information system specification [12].

3 Requirements for our framework

Based on our observations above, we now make some assumptions in order to proceed to a design for a knowledge-intensive system that supports community-based ontology evolution.

3.1 A Constructivist Approach

Humans play an important role in the interpretation and analysis of meaning during the *elicitation* and *application* of knowledge. Hence, given the diversity and the dynamics of knowledge domains that need to be accommodated, a viable ontology engineering methodology should not be based on a single, monolithic domain ontology that presumes a unique *objective* reality, that is maintained by a single knowledge engineer. It should instead take a *constructivist* approach where it supports multiple domain experts in the gradual and continuous externalisation of their *subjective* realities contingent on relevant formal community aspects [5].

Technically, this requires a knowledge engineering methodology that supports the collaborative building and managing of increasingly mature versions of *contextualised* ontological artefacts (conceptualising their divergent subject realities), and of their *interdependencies*. Ultimately, this will allow human experts to focus on the subtle "community-grounded" meaning alignment *negotiation* processes.

3.2 Modelling of Communities: Norms and Negotiation

The RENISYS method [12] conceptualises community information system specification processes as *conversations for specification* by relevant community members. It therefore uses formal *composition norms* to select the relevant community members who are to be involved in a particular conversation for specification. Next, it adopts a formal model of conversations for specification to determine the acceptable conversational moves that the selected members can make, as well as the status of their responsibilities and accomplishments at each point in time.

Similarly for our purposes, by grounding evolution processes in terms of community aspects such as composition norms and conversation modes for specification, the knowledge-intensive system can be precisely tailored to the actual needs of the community [9]. In next paragraphs, we first identify some composition norms, and then show conversation modes will play a role in meaning negotiation.

Composition Norms Among other community aspects that will orchestrate the collaborative OE processes, in this paper we only distinguish between two kinds of composition norms: (i) *external norms* that *authorise* relevant actors in the community for an action within a particular ontological context, and (ii) *internal norms* that, independently from the involved actors, *constrain* or *propagate* the evolution steps, enforced by the dependencies the involved ontological context has with other contexts.
Inspired by Stamper [29] and de Moor [12], an external norm is defined as follows:

> **if** precondition **then** actor **is** {permitted/required/obliged}
> **to** {initiate/execute/evaluate} action **in** ontological context.

The *precondition* can be a boolean, based on a green light given by an entitled decision organ, or triggered by some pattern that detects a trend or inconsistency in the actual ontological structures. The *deontic status* states whether an *actor* is permitted, obliged, or required to perform a particular *role* (initiation, execution, validation) within the scope of a certain *action* (e.g., a micro-level or macro-level OE process). A micro-level process is an operation conducted in terms of micro-level primitives, e.g. *introduceConcept*, *defineGenus*, etc. In [5], we defined such a set of OE primitives for characterising context dependencies (see Sect. 4.3).
An internal norm is defined as follows:

> {initiate/execute/evaluate} action **in** ontological context **is constrained to**
> $\{ \bigcup_i primitive_i(e_i^1, \ldots, e_i^n) \}$ **where** $\forall_i \{e_i^j, \ldots, e_i^k\} \in$ *ontological_context*$_i$
> $(1 \leq j \leq k \leq n)$.

Performing a particular action role in some ontological context is (in order to perform that action) constrained to use a restricted toolbox of primitives ($\bigcup_i primitive_i$) of which some parameters are bound to ontological elements $e_i^j, \ldots, e_i^k$, that were already grounded in some ontological contexts. In Sect. 5, we will extensively illustrate the above definitions.

Meaning Negotiation The constructivist approach engenders meaning divergence in the respective organisational contexts. This requires a complex socio-technical *meaning negotiation* process, where the meaning is aligned. However, sometimes it is not necessary (or even possible) to achieve context-independent ontological knowledge, as most ontologies used in practice assume a certain professional, social, and cultural perspective of some community. The key is to reach the *appropriate* amount of consensus on *relevant* conceptual definitions through *effective* meaning negotiation in an *efficient* manner [11]. As suggested earlier, a negotiation process is defined as a specification conversation about a concept (e.g. a process model) between selected domain experts from the stakeholding organisations. For an excellent survey on different conversation models we refer to [9].

3.3 Design for a Framework

The constructivist approach to ontology engineering in complex and dynamic realistic settings threatens to slip back into out-of-control evolution processes, when the *socio-technical* aspects are not well understood. Furthermore, these rapidly evolving community aspects, and the many dependencies they have with the actual knowledge artefacts in the knowledge structures, lead to knowledge structures that can be extremely volatile. Hence, research into a special-purpose and comprehensive framework will be needed to address the manageable evolution of knowledge structures, while respecting the autonomous yet self-organising drives inherent in the community.

We now bootstrap a design for a *community-driven knowledge-intensive system* (KIS):

1. The technical part of KIS, including a general ontology server and an API that provides collaborative *elicitation*, *representation*, and *analysis* functionalities for knowledge artefacts and context dependencies.
2. The social part of KIS, representing the client communities where communication and norms form the basis for coordinated goal-oriented action.
3. The community-grounded *meaning evolution support system* (MESS) part orchestrating the co-evolution cycle between between community communication and their knowledge.

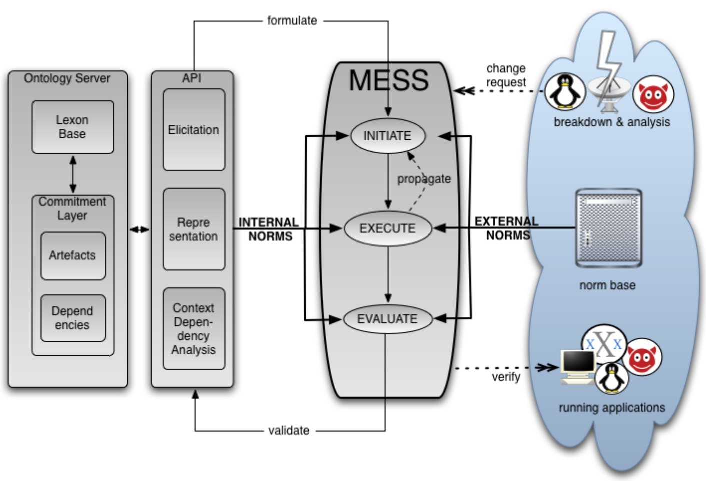

Fig. 2. A design for a knowledge-intensive system.

Figure 2 illustrates the three parts and the co-evolution cycle as follows: (i) the evolution process starts with some individual stakeholders becoming aware of a communication mismatch, that causes a work breakdown. Next, (ii) this breakdown is described in a concrete request for eliciting the relevant consensus to reinstate normal community communication. The knowledge administrator analyses and formulates the change request into concrete macro-level OE processes, which in turn can be decomposed into micro-level processes. Each of the micro-level processes engage a MESS process, which process-wise is similar to the change process model described in Sect. 2.1. Additionally, the MESS process is coordinated by the relevant participating members that are selected from the external norm base, and the impact of the changes is calculated from the formal dependencies defined by the internal norms.

For the technical part of KIS, we adopt the DOGMA ontology framework, which we present next.

4 DOGMA Ontology Engineering

Ontology is an approximate shared *semiotic* representation of a subject matter. The DOGMA [22] 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 [34].

The DOGMA approach has some key distinguishing characteristics that make it interesting for our purpose, 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 [28], is to enhance the potential for re-use and design scalability. Lexons are initially uninterpreted binary fact types, which increases their potential for reusability across community perspectives or goals. 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) [5]: (i) a taxonomical relationship (*genus*): e.g., $\langle G$, $manager$, $is\,a$, $subsumes$, $person\rangle$; or (ii) a non-taxonomical relationship (*differentia*): e.g., $\langle G$, $manager$, $directs$, $directed\,by$, $company\rangle$. 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 DOGMA characteristic 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 [13]. The meaning of the terms in a lexon is dependent on the context of elicitation [5].

E.g., consider a term “capital”. If this term was elicited from a typewriter manual (read: elicitation context), it has a different meaning (read: concept definition) than when elicited from a book on marketing. Hence, we denote:

$$concept(\langle typewritermanual, capital\rangle) \neq concept(\langle marketingbook, capital\rangle).$$

Within a context of elicitation, lexons are not merely syntactic by nature, but underspecified, what makes them reusable for being applied in a specific collaborative *application context* [34] 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.

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, 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. Ontologies can differ in syntax, semantics, and pragmatics, yet they all are built on these shared vocabularies in the Lexon Base. Examples of ontological commitments include goal, process and communication models, but also business rules, database constraints, or norms.

4.3 Context Dependency Management

Context dependencies constrain the possible relations between the entity and its context, and constrain or propagate the evolution steps within and between different ontological contexts, throughout the ontology engineering processes. Many different types of context dependencies exist, within and between ontological elements of various levels of granularity, ranging from individual concepts of definitions to full ontologies. In, [5], we formalised and illustrated three different types of context dependencies within one ontology (*intra-ontological*) and between different ontologies (*inter-ontological*): articulation (ART), application (APP), and specialisation (SPE). A typical example of a dependency type is the *specialisation dependency*, that exists between a concrete task description and a task template. In order to be complete, the task description should address the specialisation of all, and only those, *differentiae* (plural of differentia) and concepts in the template.

Context dependencies will be used to enforce internal norms. In order to constrain the applicable evolution steps, context dependencies also keep a *change log* in terms of applied micro-level primitives. Next, we will illustrate internal and external norms in real-world example.

5 Community-based Evolution of an HR Knowledge-intensive System

In order to illustrate the possibilities of our framework we consider an example that is inspired by the several research projets in the HR domain we are currently involved in[4]. Figure 3 illustrates a snapshot of the scenario, were multiple layers of ontological contexts mutually constrain each other with context dependencies.

- All ontologies commit to an extendible repository (lexon base) of *reusable competence definitions* (RCDs) (lexons). We adopt the definitions as proposed by the HR-XML consortium: an RCD is *a specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment-related characteristic (e.g. attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context*[5]. This repository also provides a set of *canonical* relationships between RCDs, including meronymical relationships, i.e. an RCD might be a *facet* or *part of* another RCD.
- The *upper interorganisational* layer includes the $SHARED_{TH}$ ontology, which defines a contributed taxonomy on RCDs.
- On the *lower interorganisational* level, several so-called (governmental) *occupational information networks* accommodate specific collaborative contexts by further "articulating" the taxonomy in $SHARED_{TH}$. For example, O*NET[6] provides a particular classification of skill RCD types. The ART (articulation) dependency between the subcontexts O^*NET_{TH} and $SHARED_{TH}$ enforces the reuse *policy* (which is an internal norm) that all RCD skill types t_{RCD} (e.g., $Basic\,Skill$) introduced in the lower O^*NET_{TH} context must be articulated by some term g_{RCD} (e.g., $Skill$) in the upper $SHARED_{TH}$ context. This policy is denoted as follows:
  execute $introduceRCD(t_{RCD})$ **in** O^*NET_{TH} **is constrained to**
  $\{$ $articulateConcept(\langle O^*NET_{TH}, t_{RCD}\rangle, c)$,
  $defineGenus(\langle O^*NET_{TH}, t_{RCD}\rangle, \langle SHARED_{TH}, g_{RCD}\rangle)$ $\}$
  where $g_{RCD} \in SHARED_{TH}$,
  where c is some concept definition. Other examples of such networks include the Flemish Social-Economical Council[7] (SERV), and the US Army Military Occupational Specialties (MOS) List[8]. They also are expected to follow this community policy when eliciting new RCD types.
- Various higher and lower level *organisational* levels (e.g., O^*NET_{RCM} in Sect. 5.2, $MOS_{Template}$ in Sect. 5.1), including branches within organisations, commit to lower or upper interorganisational levels. In the example below, $Army$ is an organisation consisting of several lower level branches[8] such as humanitarian, armor, aviation, medical service corps, etc. (not illustrated).

---

[4] http://www.codrive.org (EU Leonardo da Vinci); http://www.prolixproject.org (EU FP7 IST PROLIX); http://cvc.ehb.be/PoCeHRMOM/Frameset.htm (IWT TETRA PoCeHRMOM)
[5] http://ns.hr-xml.org/2_5/HR-XML-2_5/CPO/Competencies.html
[6] http://online.onetcenter.org/
[7] http://www.serv.be
[8] http://www.us-army-info.com/pages/branches.html

RCDs are lexically grounded and disambiguated into concept definitions, however there still are underspecified for particular pragmatic purposes. This specification happens by defining RCD maps.

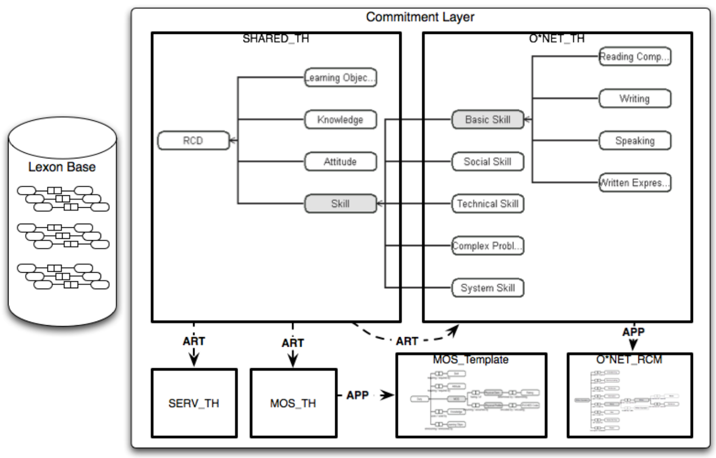

Fig. 3. A snapshot of the scenario in the DOGMA framework: on the left the lexon base, and on the right the commitment layer, were multiple levels of ontological contexts mutually constrain each other with context dependencies.

5.1 Defining RCD Maps

Organisations that commit to occupational information networks such as $O^{*}NET$, reuse RCDs and further specify (in various ways) their semantics by combining them in *reusable competency maps*[9] (RCMs), and possibly axiomatise these RCMs. Figure 4 illustrates a reusable competency map for RCD "written expression", that was extracted from the O*NET RCM subcontext "1.A.1.a.4". The APP dependency between $O^{*}NET_{TH}$ and $O^{*}NET_{RCM}$ enforces the policy that when building new RCMs for an RCD t_{RCD} (e.g., $Written\,Expression$) in the context of $O^{*}NET$, one should not introduce new RCD types, but merely reuse existing RCDs $t_2^{d_i}$ (e.g., $Understanding$) from $O^{*}NET_{TH}$ in new differentiae ($d_i = \langle O^{*}NET_{RCM}, t_{RCD}, r_1^{d_i}, r_2^{d_i}, t_2^{d_i}\rangle$). This internal norm is denoted as follows:

> execute $buildRCM(t_{RCD}, \bigcup_i d_i)$ **in** $O^{*}NET_{RCM}$ **is constrained to**
> $\{\ defineDiff(O^{*}NET_{RCM}, d_i, \langle O^{*}NET_{TH}, t_{RCD}\rangle, \langle O^{*}NET_{TH}, t_2^{d_i}\rangle\ )\ \}$
> **where** $t_{RCD}, t_2^{d_i} \in O^{*}NET_{TH}$.

Although all RCMs might be built on the same RCD base repository, they differ widely in structure and semantics, contingent on the subjective perspectives of the different organisational contexts.

[9] as defined by the HR-XML consortium

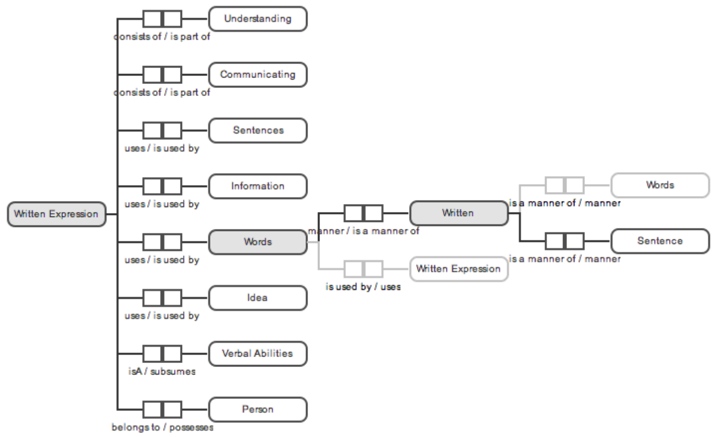

Fig. 4. RCM for "written expression" in the collaborative context O^*NET_{RCM}.

5.2 Defining Occupation Specifications

Now consider following scenario where a new military occupational specification (MOS) for "social worker" is to be introduced in the Army. The request for eliciting a new "social worker" MOS is produced in response to a breakdown in achieving a new military strategic goal towards deploying more *humanitarian operations*. These operations come in many forms, requiring HR related to confidence-building measures, power-sharing arrangements, electoral support, strengthening the rule of law, and economic and social development.

The knowledge administrator analyses and formulates the request into concrete OE processes, that are relevant to reach the appropriate amount of consensus about the new MOS in the most effective way. Two important processes are to lexically ground the term in MOS_{TH}, and to analyse the semantics of "social worker". Organisational policy requires any MOS to be *semantically analysed* according to the MOS template[10] (see Fig 5), which basically consists of two parts:

1. a general description for required skills and attitude, used knowledge, and envisioned learning objectives, specified in terms of artefacts such as upper shared RCMs or organisationally shared specifications;
2. a set of physical requirements, to be assessed with medical evidence data.

The MOS template was elicited by core domain experts and represents the current focus of the community. By specialising a MOS template in terms of reusable interorganisational RCMs it can share its call for HR, and hence attract candidates from other military organisations that have more specialised HR in humanitarian operations such as the United Nations Peacekeepers, or from civilian sectors, as the MOS is not restricted to soldiers, but also include police officers, and other civilian personnel.

Figure 6 illustrates the whole process:

[10] http://www.us-army-info.com/pages/mos/air-defense/14j.html

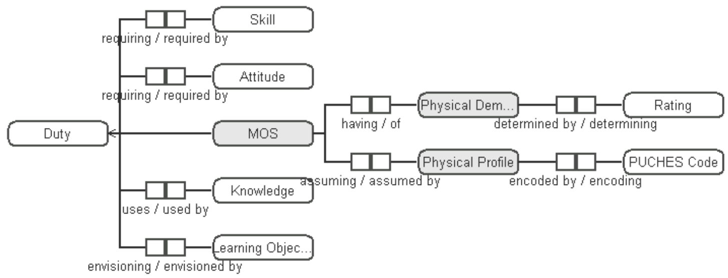

Fig. 5. A template for a military occupational specialisation (MOS) in context $MOS_{Template}$.

1. The community is aware of a collaboration breakdown, and identifies the need for a new military occupational specialisation for " social worker" as one of the solutions.
2. **initiation**:
   – this breakdown is described in a concrete change request for eliciting the relevant consensus to reinstate normal community communication.
   – if the request is accepted, the authorised knowledge administrator analyses the change request, and formulates it into concrete macro-level OE processes. Fig. 6 only illustrates this for the semantic analysis activity.
   – next, he plans the change. First, he locates the (lower organisational) collaborative context in which the analysis is to be performed, viz. $HumanOps$. Then, based on this information, he calculates the change impact. Therefore, he consults the internal norms. It turns our that in this case there are no dependent artefacts. However, as multiple members are authorised to perform each their semantic analysis of social worker, an additional negotiation process to align the resulting divergent specifications will be required.
3. **execution**:
   – once the plan is approved, it moves to the execution phase. Following norm obliges all recruiting officers (ROs) of all branches b to execute their semantic analysis activity for "social worker" in their individual subcontexts $HumanOps_{RO_b}$:
   if initialised(SemanticAnalysis) **then** $\forall_b RO_b$ **is** obliged **to** execute $SemanticAnalysis(\langle MOS_{TH}, social\,worker\rangle, HumanOps_{RO_b})$ **in** $HumanOps_{RO_b}$.
   The internal norms further constrain them to be all specialisations of the MOS template (SPE dependency between $HumanOps_{RO_i}$ and $MOS_{Template}$), and reuse RCD and RCM vocabulary from O^*NET, $SERV$ or MOS (APP dependency between $HumanOps_{RO_i}$ and O^*NET, $SERV$ and MOS). The result is a set of divergent specifications for "social worker". The execution is facilitated by providing the officers with an editing window that is precisely tailored to the job.
4. **evaluation:** The specialisations are evaluated by, e.g. defining a test population for the concepts and relationships, or by committing the organisational data schemas to them.

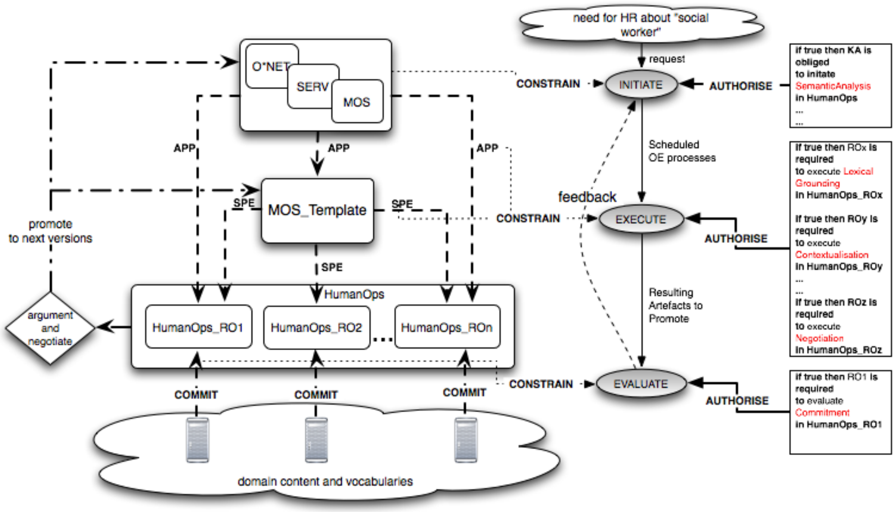

Fig. 6. Illustration of a collaborative ontology change process.

5.3 Community-based Meaning Argumentation and Negotiation

Despite the context dependencies enforcing RCD or RCM reuse and template policies, our methodology cannot exclude the possibility that policies are ignored, and hence new competency definitions are rigourously introduced ad hoc. We could further force the reuse policy by defining specific norms that would delegate the exclusive rights for defining new RCDs to the HR-XML consortium. However, such exclusive rights would be unacceptable: we have to accept that the community endorses the constructivist approach, were ontologies should be grounded in the community and in the language of the community itself. Similarly to MOS specialisations for social worker, multiple organisations (such as SERV or MOS) will have divergent RCMs for written expression, conceptualised in terms of RCDs from $SHARED$ of $O * NET$, or in terms of their own familiar organisational competency vocabulary to nuance their intensions.

The goal is that organisations can exchange their HR optimally, hence we propose a intermediate solution where organisational ontology engineering processes basically respect the policies enforced by the context dependencies, but are also allowed to introduce competencies from the organisational vocabulary. In any case where the policy is not followed, an alignment process between the stakeholding organisations should bring an acceptable balance between RCD reuse and new organisational competency vocabulary. In [4], we give a semantic account of how RCD reuse can be promoted within the DOGMA approach.

DOGMA-MESS [11] is a constructivist meaning evolution methodology and system, where such a balanced negotiation process is conducted as suggested in the requirements of KIS. In our community-grounded change process, we support DOGMA-MESS in setting up the negotiation agenda automatically: by consulting the internal and external norms, the relevant community members who are to be involved in a particular conversation, and the involved context dependencies can be selected. Ultimately, when consensus is reached, the aligned concept can be promoted and shared to the next version of the upper interorganisational level. It also works the other way around: when some consensus about an artefact is questioned after some validation period, the artefact is mandated to degrade and undergo a new negotiation round. To support the negotiation process several argumentation methods were devised such as HCOME [20] and Diligent [32].

6 Implementation

Currently a first version of a web-based DOGMA-MESS[11] is being tested in several real-world case studies, as illustrated in Sect. 5, and a client variant is being implemented in our DOGMA Studio Workbench[12] as we write. Meanwhile, we are installing norm and specification conversation models into the system, and we are planning experiments with other context dependency types (cf. [5]. In [6], we proposed a graph rewriting approach to formalise the semantics of composition norms, and conduct context dependency analysis. This approach promises to be suitable for modelling external norms as well.

7 Discussion and Conclusion

The key challenge of this article was to bootstrap a framework that studies and embraces the novel, difficult but crucial issues of adaptation of knowledge resources to their respective user communities, and *vice versa*, as a fundamental property within knowledge-intensive internet systems. By using norms to select relevant domain experts in OE processes, knowledge evolution is grounded in the community. Furthermore context dependencies, enforces organisations to reuse lower or upper shared ontologies in their local ontological contexts. However, the constructivist MESS process is also democratical in a sense that it allows organisational vocabularies to be introduced, and promoted and shared to the next version of the (upper or lower) interorganisational level. Next we discuss some observations for future research directions.

7.1 Templates

During our experiments, we experienced templates as important instruments in order to conduct knowledge elicitation in a goal-oriented way. E.g., the MOS template reflects the current shared interests regarding the specification military occupations. The template was not predefined, but also co-evolves over time with the actual community interests. In [5], we describe how template evolution triggers a cascade of changes to all its dependent specialisations. In [11] we already give some insights how new trends in the community can be detected by *relevance measures*. Based on the "wisdom of

[11] http://www.dogma-mess.org
[12] http://www.starlab.vub.ac.be/website/dogmastudio

the crowd" principle, if a certain threshold of organisations deviate from the current template, it means there is a trend shift in the knowledge elicitation process in order to serve new interests and goals.

7.2 Internal and External Norms

In this paper, we only modelled a fraction of the community aspects that play an important role in capturing community-grounded knowledge evolution. Amongst other, this will imply other context dependency types, e.g. during the evaluation phase, in order to verify the backwards compatibility of the changed knowledge artefacts with inflexible data schemas interfacing to legacy data.

7.3 Multi-disciplinary Approach

In order to better capture the communication mismatches that cause collaboration breakdown, we have to go wider than current practice by taking explorations of new and alternative approaches from multiple relevant disciplines. For example, the field of *communication modelling* and *discourse analysis* [14] has applied communication theories that are the basis for inter-organisational and inter-personal communication acts and knowledge exchange. These concepts can be used for the analysis of communication processes present in any kind of information and knowledge exchange and in particular in negotiations. Furthermore, much can be learned from the field of *information system engineering* (in particular collaborative software engineering), model-driven engineering, and *model-driven architecture* [15] offers a wealth of techniques and tools for versioning, merging and evolving artefacts. Naturally, as already mentioned, principles from the field of *organisational semiotics* can be useful in modelling communities and identifying community aspects in ontology evolution.

7.4 Human-computer Confluence

Clearly, many of the ontology engineering activities are intrinsically interactive in nature and require a lot of human intervention. This does not mean, however, that we should rule out other approaches that are fully automated. A careful balance and communication is needed between human, semi-automatic (i.e. requiring human interaction) and automatic approaches for knowledge interpretation and analysis processes. Ultimately, communities will consist of a mix of human and software agents that transparently will communicate and request services from each other in order to maintain the shared knowledge structures appropriately.

References

1. J. Banerjee and W. Kim. Semantics and implementation of schema evolution in object-oriented databases. In *ACM SIGMOD Conf., SIGMOD Record*, pages 311–322, 1987.
2. K. Bennett and V. Rajlich. Software maintenance and evolution: a roadmap. In *ICSE - Future of SE Track*, pages 73–87, 2000.

3. S. Bohner and R.S. Arnold. *Software Change Impact Analysis*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.
4. S. Christiaens, J. De Bo, and R. Verlinden. Competency model in a semantic context: Meaningful competencies (position paper). In R. Meersman, Z. Tari, and P. Herrero, editors, *OTM Workshops (2)*, volume 4278 of *Lecture Notes in Computer Science*, pages 1100–1106. Springer, 2006.
5. P. De Leenheer, A. de Moor, and R. Meersman. Context dependency management in ontology engineering: a formal approach. *LNCS Journal on Data Semantics*, 8:26–56, 2006.
6. P. De Leenheer and P. Mens. Context-driven ontology engineering: a graph rewriting approach. In *Proc. of Agtive (Kassel, Germany)*. Springer, 2007. forthcoming.
7. P. De Leenheer and T. Mens. *Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications, from Semantic Web and Beyond: Computing for Human Experience*, chapter Ontology Evolution: State of the Art and Future Directions. Springer, 2007. forthcoming.
8. A. de Moor. *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*. PhD thesis, Tilburg University, The Netherlands, 1999. ISBN 90-5668-055-2.
9. A. de Moor. Language/action meets organisational semiotics: Situating conversations with norms. *Information Systems Frontiers*, 4(3):257–272, 2002.
10. A. de Moor. Ontology-guided meaning negotiation in communities of practice. In P. Mambrey and W. Gräther, editors, *Proc. of the Workshop on the Design for Large-Scale Digital Communities at the 2nd International Conference on Communities and Technologies (C&T 2005), Milano, Italy, June 2005*, 2005.
11. A. de Moor, P. De Leenheer, and R. Meersman. DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In *In Proc. of ICCS 2006*, LNAI 4068, pages 189–203. Springer, 2006.
12. A. de Moor and H. Weigand. Formalizing the evolution of virtual communities. *Inf. Syst.*, 32(2):223–247, 2007.
13. C. Fellbaum, editor. *Wordnet, an Electronic Lexical Database*. MIT Press, 1998.
14. Grobelnik M. Fortuna, B. and D. Mladenic. System for semi-automatic ontology construction. In *In Proc. of the 3rd European Semantic Web Conf. (ESWC2006) (Budva, Montenegro)*. Springer, 2006.
15. Djuric D. Gasevic, D. and V. Devedzic. *Model Driven Architecture and Ontology Development*. Springer, 2006.
16. J. Gray. The transaction concept: Virtues and limitations (invited paper). In *Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings*, pages 144–154. IEEE Computer Society, 1981.
17. T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
18. N. Guarino. Formal ontology and information systems. In *Proc. of the 1st Int'l Conf. on Formal Ontologies in Information Systems (FOIS98) (Trento, Italy)*, pages 3–15. IOS Press, 1998.
19. R.H. Katz. Towards a unified framework for version modeling in engineering databases. *ACM Comput. Surv.*, 22(4):375–408, 1990.
20. Vouros G.A. Alonso J.P. Kotis, K. Hcome: tool-supported methodology for collaboratively devising living ontologies. In *In Proc. of the 2nd Intl Workshop on Semantic Web and Databases (SWDB 2004)(Toronto, Canada)*, LNCS 3372, pages 155–166. Springer, 2004.
21. A. Maedche, B. Motik, and L. Stojanovic. Managing multiple and distributed ontologies on the semantic web. *The VLDB Journal*, 12(4):286–302, 2003.

22. R. Meersman. The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In *Proc.of the Conf. on Cooperative Database Systems (CODAS99)*, pages 1–14. Springer, 1999.
23. T. Mens, R. Van Der Straeten, and M. D'hondt. Detecting and resolving model inconsistencies using transformation dependency analysis. In *Proc. of MoDELS/UML 2006*, LNCS 4199, pages 200–214. Springer, 2006.
24. I. Nonaka and H. Takeuchi. *The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, May 1995.
25. P. Plessers. *An approach to web-base ontology evolution*. PhD thesis, Vrije Universiteit Brussel, Brussel, 2006.
26. D. Sanderson. Cooperative and collaborative mediated research. In T. M. Harrison and T. Stephen, editors, *Computer networking and scholarly communication in the twenty-first century*, pages 95–114. State University of New York Press, 1994.
27. E. Simperl and Y. Sure. *Ontology Management for the Semantic Web, Semantic Web Services, and Business Applications, from Semantic Web and Beyond: Computing for Human Experience*, chapter The Business View: Ontology Engineering Costs. Springer, 2007. forthcoming.
28. P. Spyns, R. Meersman, and M. Jarrar. Data modelling versus ontology engineering. *SIGMOD Record*, 31(4):12–17, 2002.
29. R. Stamper. *Linguistic Instruments in Knowledge Engineering*, chapter Language and Computing in Organised Behaviour, pages 143–163. Elsevier Science Publishers, 1992.
30. L. Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, Germany, 2004.
31. L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven ontology evolution management. In *Proc. of the 13th European Conf. on Knowledge Engineering and Knowledge Management (EKAW02) (Siguenza, Spain)*, pages 285–300, 2002.
32. C. Tempich, S. Pinto, Y. Sure, and S. Staab. An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies. In *In Proc. of the 2nd European Semantic Web Conference (ESWC, 2005)*, LNCS 3532, pages 241–256. Springer, 2005.
33. C. Van Damme, M. Hepp, and K. Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *Proceedings of the ESWC Workshop "Bridging the Gap between Semantic Web and Web 2.0" (Innsbruck, Austria)*. Springer.
34. G. Zhao. Application semiotics engineering process. In F. Maurer and G. Ruhe, editors, *SEKE*, pages 354–359, 2004.