7 Knowledge-assisted Ontology-based Requirements Evolution

Smita Ghaisas, Nirav Ajmeri

Tata Research Development and Design Centre (TRDDC),

Pune, India

{smita.ghaisas, nirav.ajmeri}@tcs.com

Abstract Reaching common level of understanding of a problem domain is one of the key challenges that stakeholders face during the requirements phase of a project. The stakeholders involved in Requirements Engineering (RE) attempt to achieve this goal through communication and knowledge sharing. The process of clarifying business problems and arriving at a specification necessitates developing a common vocabulary, assigning meanings to various business concepts, determining their interrelations and reconciling stakeholders' viewpoints. Oftentimes, knowledge exists in organizations, but is not maintained in a reusable form. To address the knowledge and collaboration needs of RE stakeholders, we have developed a Knowledge-assisted Ontology-based Requirements Evolution (K-RE) method and toolset. We demonstrate creation of a knowledge repository and its reuse in two contexts: (1) to resolve Change Requests (CR) with better speed and accuracy and (2) to jump-start a new project. We combine the social software principles and semantic web concepts to achieve this.

7.1 Introduction

The aim of Requirements Engineering (RE) is to collaboratively evolve the initial uncertain and ambiguous understanding of a business problem into features and attributes of a software system. Reaching a common level of understanding of a problem domain is one of the key challenges that the software vendors and customers face during requirements definition. The process of articulating and clarifying business problems and arriving at a specification based on a shared understanding requires exchange and transfer of knowledge. This necessitates developing a common vocabulary, assigning meanings to various business concepts, determining their interrelations and reconciling multiple viewpoints from stakeholders. It also involves continually accommodating changes in the shared understanding by verifying it periodically. The challenge has become even more daunting of late because (1) software systems increasingly find applications in ever widening diversity of domains; complex and conceptually non-trivial and (2) systems are developed by globally distributed teams of stakeholders.

RE Stakeholders need knowledge from different perspectives. Customers need to see a tangible evidence of domain knowledge in an organization so that they feel confident that their requirements would be met. Requirement analysts need the domain knowledge to deliver good quality requirements efficiently [1]. Subject Matter Experts (SMEs) would like to optimize the time they spend with Requirement analysts. Oftentimes, knowledge exists in organizations and it is estimated that more than 50% of requirements knowledge for similar systems can be reused completely or with minimal modification [2]. But it is not visible and accessible easily because it may reside in a tacit form with people. Even if it is explicit in the form of documents or webpages, it is difficult to refer across the multiple disparate knowledge fragments and draw useful inferences from them. IDC report shows that the Fortune 1000 enterprises waste $5 billion annually due to intellectual rework, and the inability of finding electronic resources within the enterprise [3]. In other words, knowledge is not structured to be reusable.

Due to the distributed nature of projects, there is little or no opportunity for co-located discussions among the stakeholders. This is a threat to the success of a project. Respondents to a survey hold communication as one of the top challenges. They express that it is extremely important for the distributed teams to "use the same language" while defining requirements [4]. Empirical research reported in [5] mentions 'Lack of common language/terminology' as one of barriers to sharing the understanding of a problem domain. The success of 'social software' in achieving an effective communication has stimulated use of social principles in RE [6-8]. As a result, we see various tools weaving in collaboration into the RE process [see for example, 9]. However, although the benefits of social platforms are valuable, they are *necessary* and not *sufficient* in themselves for an effective communication. For example, the communication using the same language quoted above is not possible with just a support for collaboration, unless Requirement analysts can 'see' and access domain knowledge easily and use it to discuss requirements of their new projects. They should be able to tailor the knowledge to suit their project-specific context as well.

To address the knowledge and collaboration needs of RE stakeholders, we have developed a Knowledge-assisted Ontology-based Requirements Evolution (K-RE) method and tool. We combine the social software principles [6-8] and semantic web concepts to enable a knowledge-assisted RE [10].

The approach involves Knowledge-assisted Requirements Evolution from a generic Knowledge Base (KB). The KB consists of requirements knowledge elements such as business constraints, features, business processes, use cases, and data models. In collaboration with the customers and domain experts, Requirement analysts can modify and enhance the knowledge elements to suit project- specific needs. Each new RE exercise thus becomes a guided evolution of a generic KB so that it meets project-specific needs. This is characteristically different from the traditional 'clean slate' RE approach; hence the term Knowledge-assisted Requirements Evolution (K-RE). The just-in-time, context-sensitive assistance based on semantic web ontologies, knowledge elements and inference rules operating on them serves as a guidance and moderation mechanism. This complements the collaborative identification, discussion and definition of requirements enabled by the underlying social platform. K-RE has three roles- Requirement analysts who consume knowledge, Knowledge contributors who capture and structure domain knowledge, and Knowledge curators who review contributions made by other experts to maintain the currency and correctness of knowledge.

We have deployed K-RE in three pilots and have evaluated its usefulness. In this paper, we present the results of applying K-RE to demonstrate its benefits in the context of knowledge reuse in a large distributed Insurance project.

By weaving in domain knowledge seamlessly into an RE process, we respond directly to the call of RE community [11] for a continued research into RE process improvement. Both, -the well documented high cost of requirements related problems [12-15] and the benefits of improvements in RE processes [16] serve as a motivation for our work.

The remainder of this paper is organized as follows. In Section 7.2, we describe how social platforms and semantic web influence RE. Section 7.3 details the solution approach and underlying model. Section 7.4 describes the process of knowledge representation and reuse. Section 5 details K-RE tool and illustrates its usage. In Section 7.6, we present the results of deploying K-RE in two different industrial settings. Section 7.7 presents the conclusion and Section 7.8 talks about the future work.

7.2 Social Platforms, semantic web and RE

Social platforms and the semantic web [17, 18] have influenced software engineering significantly [6-8]. The emerging social software engineering discipline is about enabling community-driven creation, management and deployment of software by applying methods, processes and tools in online environments [6, 19-21]. In this section, we highlight some relevant characteristics of the two paradigms in the context of knowledge reuse in RE. We address issues such as self-organization versus moderation, democratic voting versus weighted voting, bottom-up folksonomies versus top-down taxonomies and semantically enriched, hence more meaningful and effective collaboration.

The social nature of web 2.0 has been credited with democratizing knowledge content. The same democratic aspect also assigns ownership and responsibility to the content creators. RE stakeholders can use the democracy to identify, debate, and define requirements collaboratively and determine what parts of existing domain knowledge can be reused. The highly transparent communication that web 2.0 platforms enable can contribute constructively to exchange of ideas and healthy criticisms. A wiki-like platform for entering, editing requirements and for deliberating them openly is therefore highly suitable for RE.

However a platform meant for RE is likely to benefit from supervision and moderation by requirements experts. Lohmann et al [6] note that supervision and moderation by requirements experts remains crucial to a project's success. They emphasize that the moderation should be unobtrusive This seemingly defeats the purpose of a 'social' platform, but the comments and discussions are for all to see, and no single heavy-weight stakeholder can unfairly overrule valid suggestions made by even junior stakeholders, or they would face pressure from other experts in the community. Such a moderation can be in the form of a context–specific semantic assistance built into the RE process. An underlying knowledge framework of semantic web ontologies and inference rules can be employed to select relevant parts of domain knowledge for reuse and provide just-in-time context-specific suggestions to collectively evolve a requirement specification.

Social platforms provide for voting mechanisms. A social platform meant for RE however should take into account the opinions of stakeholders such as domain experts with a higher weight than less experienced stakeholders in an organization. Roles that incorporate suitable weights for respective stakeholders can be useful for this purpose. For example, a 'Knowledge contributor' role has a higher weight than a 'Requirement analyst' role, but a lower weight than a 'Knowledge curator' role. We note here that this approach is intended to ensure a meritocratic treatment of participants' opinion. Hence, it is the responsibility of project managers to assign higher weights to the opinions expressed by knowledgeable participants such as domain experts. If instead, the weights are decided based on hierarchy alone and without making knowledge the central criterion for a higher weight, the meritocratic purpose will be obviously defeated.

Folksonomies [22] that result out of free-form tagging have emerged as a way of organizing knowledge on social platforms. RE stakeholders can arrive at a shared understanding of a domain using this mechanism. Folksonomies are easy to create and hence popular, but they lack structure. Also, if some concepts are not tagged using the right term, it is hard to search and detect them. A collaborative RE platform however cannot be entirely free of structure. A foundational structure in the form of a predefined taxonomy of RE concepts and domain-specific terminology can be valuable to its stakeholders in arriving at a shared understanding of a domain. Providing a user interface that lets one use predefined concepts and add new concepts easily should strike the right balance between ease of use associated with free form tagging and rigor associated with structure, semantic precision and synonym control. Semantic web ontologies [26, 27] are a way of rendering such a structure to the platform.

The social software and semantic web complement each other and can enhance the effectiveness of knowledge reuse in an RE process. Ankolekar et al [7] emphasize that in fact both the streams need elements from the other to overcome respective limitations. They hold that semantic technologies bear a great potential of providing a robust and extensible basis for web 2.0 applications.

The social software platforms, through their democratic spirit serve to connect humans (e.g. identifying experts in the RE community), whereas the semantic web ontologies provide a mechanism to assign unambiguous meaning to vocabularies and link them. Using semantic web ontologies, we can assign distinct, persistent URIs to each term and relationship (e.g. Insurance domain specific concepts and their relationships). Therefore, linking them with each other and with other web resources is easy. Social platforms and semantic web thus provide distinct and yet complementary network effects. Combining the two paradigms in the context of knowledge reuse in RE is a compelling way to increase their total value significantly.

7.3 Foundations of K-RE

As discussed earlier, while adopting the social software principles to a highly specialized field such as RE and requirements knowledge reuse, we need to complement the social aspects with semantics [6, 7].

K-RE organizes knowledge along four distinct contexts (1) Environment (2) Problem domain (3) Generic Requirements (4) RE Process. The semantic assistance in K-RE comes from the inference rules operating on the four ontologies that represent these knowledge contexts. The framework also incorporate abstractions from various knowledge modeling paradigms like feature models [23], business process models [24], data models and use case models [25], to capture and organize knowledge elements.

7.3.1 Solution architecture: Ontologies as an underpinning framework for requirements

The four ontologies in K-RE, - 'Environmental Context Ontology', 'Generic Requirements Ontology', 'RE Process Ontology' and 'Problem Domain Ontology' are created using RDF-OWL schema [26, 27]. Fig. 7.1 shows example instances of the ontologies depicted using the UML class diagram notation.

7.3.1.1 Environmental Context Ontology

This ontology is designed to capture the environment for which software requirements are to be defined. For example, a Requirement analyst may want to capture requirements for a ***Business Process*** *–Member Enrolment of a Pension* application for a ***Customer*** *Europe Company1* in the *Europe Country1* ***Geography***. The abstractions ***Actor***, ***Action***, ***Domain***, ***Line of Business***, ***Customer***, ***Geography*** are used to capture the information.

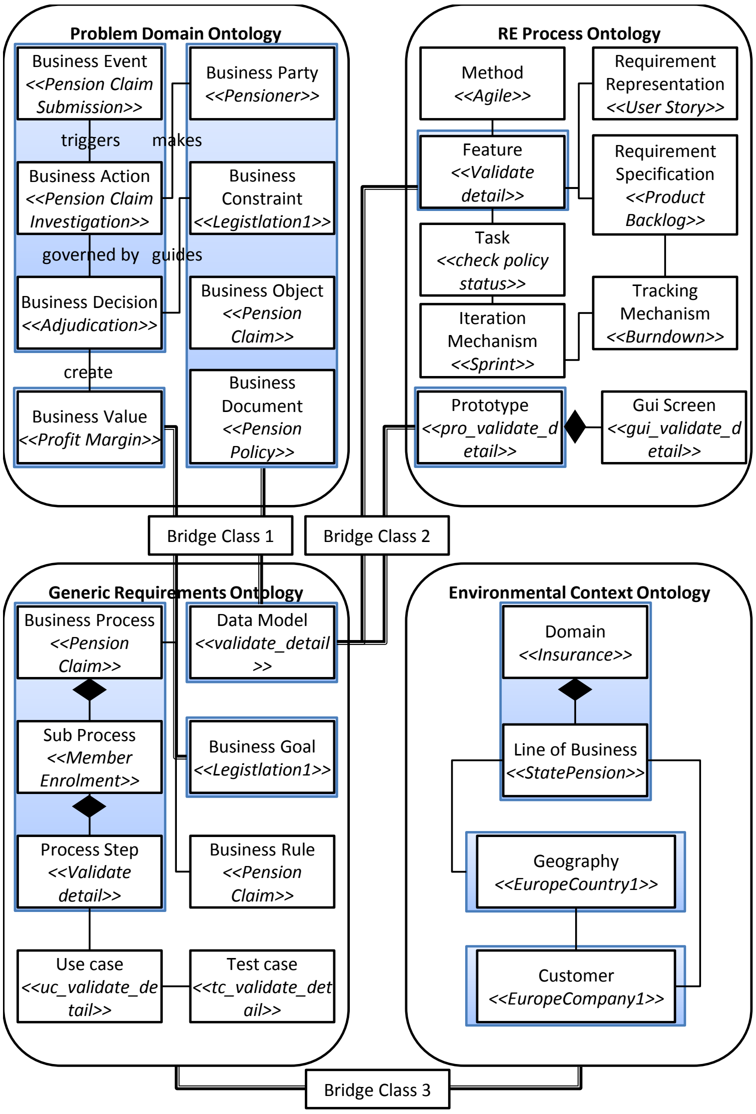

Fig. 7.1. Example KB instances and bridge classes that enable context-specific recommendations

7.3.1.2 Problem Domain Ontology

This ontology provides abstractions to capture the essence of the problem domain. For example, consider the following scenario- *'In event of death of a pensioner, a beneficiary may submit a claim request'*. The abstractions such as ***BusinessEvent***, ***BusinessType***, ***Party***, and ***BusinessAction*** are used to capture this information.

7.3.1.3 Generic Requirements Ontology

The KB that we present to the Requirement analyst is created in terms of requirements knowledge elements such as ***business goals***, ***features***, ***business processes*** and ***sub-processes***, ***business constraints*** (laws of the land, organizational policies), ***use cases*** and ***business entities***. The Generic Requirements Ontology provides these abstractions [28 and references therein].

7.3.1.4 RE Process Ontology

This contains abstractions specific to the RE process, for e.g. *Agile* ***Method*** has ***Requirement Representation*** in form of *User Story*, ***Iteration Mechanism*** as *Sprint* and ***Tracking Mechanism*** as *Burndown*.

7.3.2 Examples of mappings between the elements of different ontologies

- The ***BusinessEvents*** (e.g. *Pension Claim Submission*), ***BusinessActions*** (e.g. *Investigate Pension Claim*) and ***BusinessDecisions*** (e.g. *Adjugation*) in the **Problem Domain Ontology** are represented as ***BusinessProcess*** (e.g. *Pension Handling*) in the **Generic Requirements Ontology**.
- ***BusinessConstraint*** (e.g. a *New legislation#123*) in the **Problem Domain Ontology** in maps to ***Validation*** (e.g. *Verify conformance to legislation#123*) in **Generic Requirements Ontology**.
- The ***BusinessParty*** (e.g. *Member*), ***BusinessObject*** (e.g. *Pension Claim*), ***BusinessDocument*** (e.g. *Pension Policy*) from the **Problem Domain Ontology** contribute to ***DataElement*** in the **Generic Requirements Ontology**.
- ***Feature*** (e.g. *Validate Member Details*) in **RE Process Ontology** maps to ***SubProcess*** (*Member Enrolment*) in **Generic Requirements Ontology**.

The semantic assistance is achieved by employing the 'Bridge classes' and inference rules written in the Semantic Web Rule Language (SWRL) [29]. The Bridge classes specify semantic mappings between the four ontologies. We define rules that refer the ontology-instances and provide recommendations based on the integrated inference. The recommendations may be specific to a singular ontology or may span across the four ontologies when necessary; in response to actions of the Requirement analyst. For example, if a Requirement analyst selects *Europe Country1* as the ***geography*** for a *Pension application* to be developed, she would be presented with ***features*** and *user stories* relevant to *Insurance Pension* ***Processes*** from the KB. As she configures them in the context of her project, she would be presented with ***business rules***, in the given ***geography*** e.g. *Pension rules* in *Europe Country1* (**Environmental Context Ontology** and **Problem domain Ontol-**

ogy). If she selects ***features*** that complement each other but decides to associate them with different *sprints*, she would receive a recommendation to preferably re-arrange them in the same *sprint* (**Problem Domain Ontology** and **RE Process Ontology**). If she adds a new ***feature*** to the *Product backlog* upon the Customer's suggestion, and it happens to conflict with an already selected ***feature*** in a given domain, she would be alerted about the inconsistency of her selection (singularly the **Problem Domain Ontology**). Requirement analysts can improve the completeness, correctness and consistency of requirements using the in-built domain knowledge served to them in the form of just-in-time alerts.

Using the collaboration mechanisms in K-RE, dispersed teams can interact informally. Moreover, K-RE provides for a semantically enriched collaboration that would foster meaningful and focused discussions on topics in requirements engineering in general and problem domain such as Insurance in particular. The stakeholders can even evolve the ontologies collaboratively. A user is at liberty to identify new concepts as and when necessary. This constitutes the folksonomy which evolves bottom-up in a community-driven way. If their usage in the community of practice (in this case, RE stakeholders) is substantial, the concepts can be absorbed into the taxonomy. This decision can be made collaboratively and with advice from domain experts.

7.4 Process for Knowledge Representation and Reuse

In this section, we illustrate capturing of knowledge from requirements documents and web sites and mapping the knowledge elements onto the K-RE model.

7.4.1 Knowledge from documents

Knowledge representation is done in three steps: (1) Identification of structural details of sources such as requirements documents and web sites (2) Mapping of the document structure to the K-RE model (3) Extraction of domain knowledge from documents and its representation in K-RE.

7.4.1.1 Identification of structural details

Each project has its own set of templates to organize requirements knowledge. In this step, we identify the details of the document structure such as headings, sections and subsections. For example, a document may contain a section to list business rules. We note that this section needs to be mapped to ***BusinessConstraint*** in the K-RE model in the next step. We use the K-RE model as a reference to identify missing structure details and knowledge elements in the requirements document. For example, K-RE model has a notion of ***SystemUsecase***, which may not be present in a project's requirements specification.

7.4.1.2 Mapping of document structure to K-RE model

Knowledge contributor defines the scope of the KB by setting the environment parameters (e.g. ***Domain***- *Insurance*, ***Line of Business***- *State Pension*, ***Geography*** *Europe Country1*, ***Customer***- *EuopreCompany1*). The Knowledge contributor maps the structural details identified in the previous step to various concepts in the K-RE model. For instance, ***Sub-process*** or ***Functionality*** in requirement documents maps to the concept ***Feature*** in K-RE. This is a semi-automated and semantically-assisted process; points of human intervention are identified explicitly. K-RE prompts for inputs from users where necessary. For example, if a subsection such as ***Overrider*** in a ***use case*** cannot be mapped to any concept in the model, K-RE detects that it is unmapped and the user is asked to either map it to some existing concept or define a new concept that accommodates the detail. The structural details thus extracted correspond to the Generic Requirements Ontology (Section 7.3.1).

7.4.1.3 Domain knowledge extraction and representation

We have observed the following common problems in requirements documents.

- Granularity of domain knowledge elements such as business processes and, use cases is unspecific.
- Business constraints are not explicitly documented; they are often embedded within the business processes-e.g.in alternate flows.
- Business process steps are not always associated explicitly with actors who perform them.
- Manual business process steps are not explicitly differentiated from the automatable ones
- There is no uniformity in the usage of business terms.

K-RE aims to reduce these ambiguities by providing well-defined abstractions using the underlying ontological model and a semantic assistance to organize domain knowledge.

K-RE detects terms and key phrases specific to a given domain (such as Insurance in our case). The detected key phrases are instances of concepts in the ontologies. For example, a term such as *Pension Policy* is a concept- instance of ***BusinessDocument*** in the Problem Domain Ontology. The Knowledge contributor is presented with the concepts from the Problem Domain Ontology and is required to map the concept-instances extracted from the document to the concepts in the ontology. The identified concept-instances are then parsed to detect similarity mappings. The techniques employed are lexical similarity [30], semantic similarity

[31], direct string matching and ontological structure based mapping [26, 27]. If concept- instances that do not exactly correspond to any existing concepts in the ontology are identified, the Knowledge contributor can define new concepts or reconcile them with the existing concepts if possible.

The concept-instances are also parsed to detect associations between concept-instances (e.g., *Member participates in Scheme*). Relations like subclass, super class, equivalent, part-of, and concepts related with each other by minimum cardinality of one on both sides are considered. The identified concept- instances and associations between them are subject to refinements by human intervention.

If a key phrase is identified as a feature, the Knowledge contributor is asked to specify complementary feature(s) and conflicting feature(s) from a list of available features in the existing KB. She can also add new complementary/ conflicting features to the KB. (e.g. *Retirement Benefit Option to Purchase Annuity* is followed by *Decumulation (disinvestment of funds)*). The abstractions to capture complementary and conflicting features take cues from 'requires' and 'excludes' relationships in feature modeling [23].

If a key phrase is identified as a use case, the Knowledge contributor specifies actor(s) from the available list or adds new ones to the KB. She also identified 'includes' and 'extends' use cases for a given use case from the KB or adds new ones. Each sentence is processed and all sentences in passive voice are converted to active voice. We use triplet extraction to identify part of speech [32] in form of Subject-Predicate-Object. The subject refers to the performing actor and the predicate and the object pair in the verb phrase refers to the use case. These abstractions are captured as per the use case model [25].

Business constraints (e.g. *'Member's Benefit Age should be the age notified by member to the Scheme Administrator'*) typically contain terms such as ***should, must, if, if-else, only-if.*** Key phrases containing these terms are presented to the Knowledge contributor so that she can map them as instances of ***BusinessConstraints***. Whenever a business constraint is identified, the features, the use cases that are affected by the constraint are specified. Complementary and conflicting constraints are specified. The constraints are then classified into ***Rules*** that are domain-specific, ***Laws*** that are locale-specific and ***Policies*** that are company-specific.

The domain knowledge elements thus extracted map to the Problem Domain Ontology (Section 7.3.1) and other knowledge models. The Knowledge curator validates the correctness and currency of the KB created using the process.

7.4.2 Knowledge from domain-rich web sources

Organizations sometimes venture into new domains and need to build KBs from external sources. One of the prominent external sources today is the web. To accumulate knowledge from the web, K-RE employs a web crawler developed in-house to explore various resources. We do a targeted crawling for websites whose patterns are known. If the pattern or template of the website is not known (or if the web source does not follow a fixed template), we prune the html tags and parse only the text to extract concept- instances. The mapping of the web page pattern to K-RE model and the process of identifying the concept-instances, associations, features, use cases and business constraints, and validation by Knowledge curator remains the same as the one described in Section 7.4.1.

7.4.3 Knowledge Reuse and Upkeep

The structured KB enables reuse of knowledge while defining requirements of a new application in the same domain. A suitable instance of KB is made available to Requirement analyst as per the project environment.

A Requirement analyst who wants to work on a new Pension project starts with selecting environmental parameters. Based on the selected environmental parameters she is presented with a core set of features from the KB that matches the parameter selection (e.g. *State Pension*). As she proceeds to select the relevant features, she is recommended to include the complementary features and avoid the conflicting ones. Based on the feature selection she is presented with the relevant business rules, business glossary associated with the selected feature.

If Requirement analysts learn some new information about a domain, they can add/edit/remove the knowledge elements from the repository as well. K-RE identifies the usage patterns of the Requirement analysts and makes them visible to the Knowledge curator.

7.5 K-RE - A Tool for Knowledge-assisted collaborative Requirements Evolution

In this section, we describe the tool for knowledge-assisted requirements evolution. We also illustrate its usage with an example.

7.5.1 Overview

The tool provides a wiki-like user interface for Knowledge contributor and Knowledge curator to contribute domain knowledge, and for Requirement analyst to access and configure the knowledge.

- **Knowledge contributor:** An experienced Requirement analyst who contributes generic and specific domain knowledge to the repository.

- **Knowledge curator:** A subject matter expert, the curator monitors and ensures quality of knowledge. She ensures that the knowledge content is correct and current. Curator acts as the reviewer and monitors the contributions made by the Knowledge contributor.
- **Requirement analyst:** End user of the K-RE who configures available domain knowledge as per the Project Environment parameters and scope.

7.5.2 Architecture

K-RE is a web-based tool with a centralized application server and database server accessible to multiple clients over the internet. To store knowledge models and their instances, K-RE uses RDF-OWL [26, 27] ontologies and a relational datastore [33]. The design incorporates collaborative aspects of web 2.0 for a participatory information sharing and collaboration among the RE stakeholders. The semantic assistance provided by K-RE uses OpenNLP – NLP toolkit [34], WordNet lexical database [35] and RDF-OWL ontologies along with SWRL rules.

Fig. 7.2 shows the tool architecture. The presentation layer serves as the interface between the tool and the user. The logic layer incorporates a Guidance Enabler and Content Processor. The user requests, sent using the presentation layer, are processed by the logic layer. The data layer includes the knowledge reference module to represent knowledge in form of ontology concepts and instances.

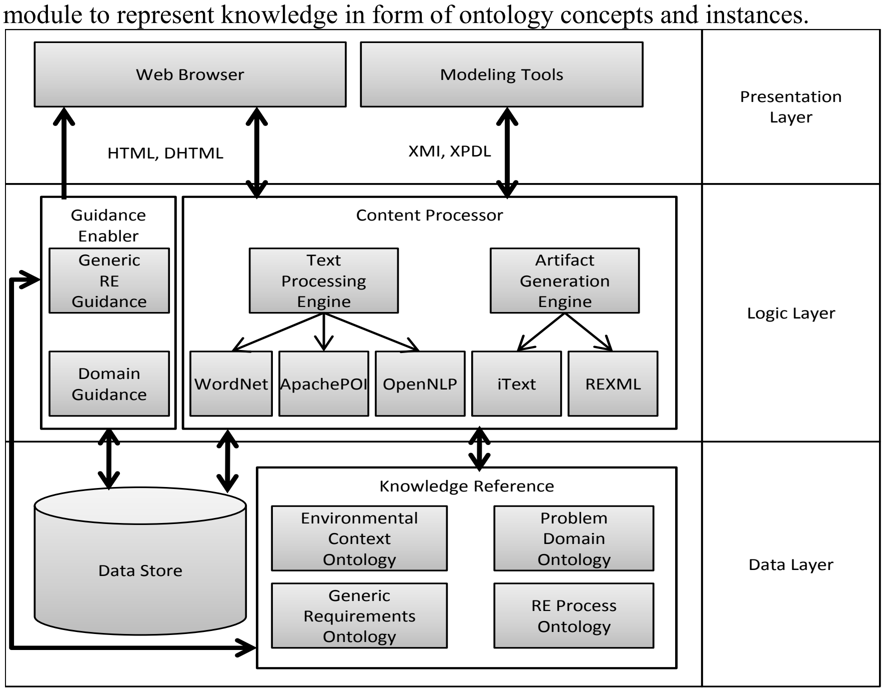

Fig. 7.2. K-RE Architecture

7.5.3 Usage Illustration

Table 7.1 shows K-RE activities and knowledge assistance with examples.

Table 7.1. K-RE activities and knowledge assistance

K-RE Activities	**Knowledge assistance from '*State Pension*' KB**	**Example(s)**	**Formalized/ Human Intervention**
Select environmental parameter	A KB relevant to the selected parameters is presented	Parameters: ***Domain*** (e.g. *Insurance*), ***Line of business*** (e.g. *State Pension*), ***Geography*** (e.g. *Europe Country1*) and ***Customer*** (e.g. *EuropeCompany1*), KB including ***Processes*** such as *Member Enrolment, Contribution, Alterations, Settlement & Exits* presented.	Formalized
Select Feature	Domain specific recommendation of complementary and conflicting nature of the features	Selected ***Feature***: (1) *Early Retirement due to serious ill health*, (2) *Over Maximum Retirement Age Processing*. Recommendation to reconsider the selection due to conflicting nature of features. Underlying ***Business Rule***: *A member is eligible for early retirement due to serious ill-health before the Maximum Retirement Age.*	Formalized
Edit Feature	Domain and geography specific recommendation to include relevant Business Rules/Policies, Business Terms	Selected ***Feature***: *Early Retirement due to serious ill health* ***Business Term***: *Deferred Retirement* ***Rule***: *Laws of land related to Early Retirement*	Formalized
Edit Feature Step	Domain/ Geographic/ Context specific recommendation for most agreed upon Business Term	Altering ***Feature Step***: *Check if retirement age is less than Normal Benefit Age* Most Agreed Upon Term: *Normal Retirement Age* Synonymous Term: *Normal Pension Age, Normal Benefit Age*	Formalized+ Human Intervention
Specify interactions with external domain(s)	Possible interactions with other domains are presented	Requirement analyst can view possible interactions with other domains and identify entry points into relevant processes. ***Feature***: Early Retirement due to ill health Possible *Interfaces* with External Domain(s): *Healthcare Domain* (e.g. *Reviewing Member's Medical Evidences submitted for Serious ill health to confirm life expectancy of member is less than a year.*) *Banking Domain* (e.g. *Payout Processing after Early retirement claim is approved*)	Formalized+ Human Intervention

In addition to those illustrated in Table 7.1, K-RE enables the following:

7.5.3.1 Semantically enabled collaboration

While carrying out any of the activities in Table 7.1, a Requirement analyst can start discussions in the form of informal chats on the selected knowledge elements with her colleagues, experts and seek opinion on selections from the KB and refinements to be made. She can post topics for discussions on semantically enabled forums and subscribe to alerts when others post their opinions on topic of her interest.

For example, if she selects the following rule to be included in her specification: '*Member's Benefit Age should be the age notified by member to the Scheme Administrator*' and is not sure if this is valid in *Europe Country2*, she can start a forum to discuss this with experts. Upon initiating a forum, she will be presented with a set of relevant posts available on the topic. For example, she can view posts related to validity of rules for *Pension*, rules for *Europe Company1*, rules for *Europe Country2,* posts by other experts who contributed Pension rules, rules regarding related terms such as date of commencement premium and select the most suitable thread of discussions in terms of topic, author geography and so on.

7.5.3.2 Generating and refining artifacts iteratively

The Requirement analyst can generate structured requirements specification documents intermittently. If project follows an *agile* method, she can view *Sprints, Product backlogs* and *Burndown charts*. She can also populate ***data models*** using modeling tools. The analyst can either work on the 'text' or 'diagram' and import/ export to /from either format. This helps in refining artifacts incrementally.

Starting with the KB for business process *Member Enrolment*, the Requirement analyst can thus evolve a specification that suits a given project. The evolution is an assisted exercise that helps in adding to or modifying the KB by providing context-sensitive help to a Requirement analyst.

It is relevant here to add that not all of the domain knowledge is formalizable in terms of ontologies and the semantic web rule language (SWRL). We therefore use a combination of formalization and human intervention to represent knowledge and enable its reuse in RE. The example presented in Table 7.1 indicates points of human intervention.

7.6 Evaluation in Industrial settings

In this section we present results of deploying K-RE in a large Insurance project. The project under discussion is a large Europe Country1 workplace pension scheme. It involves providing scheme administration services to the client.

We illustrate reuse of the structured knowledge in two contexts: (1) Change impact analysis in the same project (*Europe Country1*) and (2) while starting a new project (*Europe Country2*) in the same domain.

We created a *State Pension* KB using K-RE from existing requirement specifications. Table 7.2 lists the tasks and effort involved in creating the KB.

Table 7.2. Tasks and effort involved in creating knowledge base

Task	Effort
Analysis and standardization of requirement specification documents	10 person days
Uploading in K-RE	1 person day
Review of extracted knowledge elements	8 person days

The details pertaining to the size and content of the KB are given in Table 7.3. In parentheses against the knowledge elements, we indicate the number of the respective knowledge elements identified in the documents.

Table 7.3. Knowledge Elements from project documents and their mappings to the K-RE model

Requirement Artifacts identified from Project Documents	Problem Area	Mapping details	Corresponding Knowledge Elements in K-RE
Group (5)	Multiple Business Processes within a group	Individual processes (e.g. Settlement and Exits, Fund Administration) from a group (e.g. Group 4) were extracted and mapped to a Process in K-RE	Process (16)
Sub Group (13)	Multiple Business Sub-Processes within a sub group	Individual sub-processes (e.g. Retirement, Transfers, Death and Cessation) from a sub group (e.g. Settlement and Exits) were extracted and mapped to a Sub Process in K-RE.	Sub Process (52)
Functional Use case (48)	Multiple functionalities within a Use case	Piece of Business Functionality (e.g. Early Retirement due to ill health, Early Retirement due to Incapacity) that can execute separately were extracted from the Use cases and Business Flows, and mapped to a Feature in K-RE. (Pieces of Functionalities from 48 Functional use cases and 333 Business Flows were mapped as 220 Features)	Feature (220)
Business Flows (333)			

Business flow step	Actors not associated with each business flow step	Each feature step was associated with a performing actor.	Feature Step
Business Rules(218)	Many business rules embedded in the Business flows	128 additional business rules were identified from the business flows through domain analysis.	Business Rule (346)
Glossary Terms (517)	Inconsistency in the usage of business terms	Additional business terms detected from documents. Business terms and the relation between them were identified.	Business Terms (1210)

The KB was found to be useful in two ways- (1) For resolution of change requests in the same project and (2) As a point of departure for a new Pension project.

7.6.1 Change Request resolution in the same project

For each proposed requirement change request, the Requirement analysts carry out the change impact analysis and discuss with other stakeholders, the effort involved in implementing the change. Changes in requirements always have a ripple effect [36]. Bohner [37] defines Change Impact Analysis (CIA) as "the activity of identifying the potential consequences, including side effects and ripple effects, of a change, or estimating what needs to be modified to accomplish a change before it has been made". The effort involved in implementing a change is proportional to the impact and has a direct bearing on cost of the project. The process of change resolution thus benefits from the (1) A visible knowledge about impacted requirements elements as a result of the change in a given element and (2) Collaboration mechanisms that allow for discussions among Requirement analysts, developers, project managers and customers. K-RE incorporates both these aspects.

To evaluate the usefulness of K-RE, we conducted an experiment to compare the manual change request implementation routinely practiced by the project and one carried out using K-RE. Two separate groups of Requirement analysts implemented 30 change requests for the project. (1) Group 1 consisting of seven Requirement analysts followed the traditional approach of analyzing the requirement specification documents to identify impacted elements and (2) Group 2 consisting eight Requirement analysts implemented the changes using K-RE. The change request implementation was reviewed by five domain experts.

7.6.1.1Change Request Analysis

Table 7.4 shows an example of the original requirement and the change request.

Table 7.4. Requirement Change Request details

Requirement Details	
Original Requirement	Introduction of a Default Retirement Age: Member's Benefit Age should default to Nominal Benefit Age/State Pension Age.
Change Request	Introduction of two Default Retirements Age: 1(a): Revise the rule of Nominal Benefit Age to automatically adjust the Benefit Age of member. Member's Benefit Age should default to Nominal Benefit Age/State Pension Age. Incase member does not take their benefit and do not tell when they intend to take their benefit, then after the expiration of Nominal Benefit Age/State Pension Age, the nominal benefit age should automatically default to one day before the Maximum Retirement Age (i.e. 75) 1(b): Change the current definition of 'Nominal Benefit Age' and replace the term with 'Benefit Age

The proposed requirement change implicitly includes change in *Nominal Benefit Age* related ***business rules*** and change in the use of some ***business terms***.

7.6.1.2 Traditional Change Request Resolution

Requirement analysts in Group 1 performed the change impact analysis and determined the related requirement artifacts. They used the traceability information available in requirement specification documents.

We noted the following:

- *Total number of impacted knowledge elements identified by Domain Expert (IKE $_{Total}$):* These are the total number of knowledge elements impacted by the proposed change assuming all knowledge elements have been captured.
- *Total number of impacted knowledge elements identified by Requirement analyst manually (IKE $_{Manual}$):* These are actual number of impacted knowledge elements identified by the Requirement analyst manually from requirement specification documents.

7.6.1.3 Change Request Resolution using K-RE

Analysts in Group 2 used K-RE to handle change requests. Using K-RE, they modified the ***business rules*** corresponding to *Nominal Benefit Age* in the ***business process*** *Settlements & Exits* and ***sub process*** *Retirement*. Upon making these modifications, they received alerts to also update associated knowledge elements such as complementary ***rules***, and ***validations***. When the ***validations*** were updated, K-RE prompted the analysts to review corresponding ***task*** and ***feature step***. For each ***feature step*** altered, K-RE presented all the knowledge elements associated with it. Table 7.5 illustrates some examples for change request under consideration.

Table 7.5. Illustration of Requirement Change Resolution using K-RE

Requirement Change Resolution Activities	**Guidance from K-RE**	**Identifying Impacted Knowledge Element**
Update Business Rule	Domain specific recommendation to update Complementary Business Rule	Rule#1: "Member's Benefit Age should be the age notified by member to the Scheme" Complementary Rules Rule#2 "In the absence of Member notification, Member's Benefit Age should be State Pension Age" Rule#3 "In the absence of Member notification, if member has already attained relevant age then Member's Benefit Age should be immediately before the Member attains age 65."
	Generic recommendation for requirements completeness to update Corresponding Validations	Validation#1: Check if Member Age is less than 65
Update Validation	Generic recommendation for requirements completeness to update corresponding Task	Task#1: Validate member age with Normal Minimum Retirement Age
Update Task & Feature Step	Generic recommendation for requirements completeness about possible impacted knowledge elements associated with Feature step	Associated Actor , Associated Triggering Feature Step,Associated Nested Feature Step,Associated System Use case

We noted the following:

- *Total number of impacted knowledge elements identified by K-RE (IKE $_{K\text{-}RE}$):* These are the total number knowledge elements identified by K-RE as impact of the proposed change.
- *Total number relevant impacted knowledge elements identified by K-RE (IKE $_{K\text{-}RE\ Rel}$):* These are the impacted knowledge elements that were identified by K-RE and considered relevant by the domain expert.

7.6.1.4 Effectiveness Parameters

In order to measure the effectiveness of requirement change resolution using K-RE and compare it with to the traditional approach, we computed Precision and Recall values based on the data obtained from the experiment.

Precision: We define Precision as, 'Percentage ratio of relevant impacted knowledge elements identified that require change to the total impacted knowledge elements identified'.

Precision (P_M) is the percentage ratio of impacted knowledge elements identified by the Requirement analysts in Group 1 that were considered relevant by the domain expert (to the total number of impacted knowledge elements identified by the Requirement analyst.)

Precision ($P_{K\text{-}RE}$) is the percentage ratio of impacted knowledge elements identified by the Requirement analysts using K-RE that were considered relevant by the domain expert (to the total knowledge elements identified by K-RE).

$$P_{K\text{-}RE} = (IKE_{K\text{-}RE\ Rel} / IKE_{K\text{-}RE}) \times 100$$

Here, we have assumed that Precision (P_M) is 100% because all the elements that the Requirement analysts identify are correct. However, Requirement analyst may or may not identify all the impacted knowledge elements that require change. We have verified this logical assumption with domain experts.

Recall: We define Recall as, 'Percentage ratio of impacted knowledge elements identified to the total actual impacted knowledge elements'.

Recall (R_M) is the percentage ratio of the relevant knowledge elements, identified by the Requirement analyst, to the total actual impacted knowledge elements.

Recall ($R_{K\text{-}RE}$) is the percentage ratio of the relevant knowledge elements, identified by K-RE to the total actual impacted knowledge elements.

$$R_M = (IKE_{Manual} / IKE_{Total}) \times 100$$
$$R_{K\text{-}RE} = (IKE_{K\text{-}RE\ Rel} / IKE_{Total}) \times 100$$

7.6.1.5 Results

Table 7.6 lists the parameters discussed earlier.

Table 7.6. Impact analysis of Change Requests

Change Request	**IKE_{Total}**	**$IKE_{K\text{-}RE}$**	**$IKE_{K\text{-}RE\ Rel}$**	**IKE_{Manual}**
CR 1	4	5	4	2
CR 2	6	7	6	6
CR 3	11	12	11	5
CR 4	10	10	9	4
CR 5	10	10	9	4
CR 6	13	11	11	7
CR 7	12	11	9	6
Total	$\sum IKE_{Total} = 154$	$\sum IKE_{K\text{-}RE} = 152$	$\sum IKE_{K\text{-}RE\ Rel} = 138$	$\sum IKE_{manual} = 80$

The plot in Fig. 7.3 depicts the Precision and Recall value computed for each of the CR handled manually and using K-RE.

Precision -

Average Precision of the K-RE was computed as *90.79%.*

$P_{K\text{-}RE} = (\sum IKE_{K\text{-}RE\ Rel} / \sum IKE_{K\text{-}RE}) \times 100$

$P_{K\text{-}RE} = 90.79\%$.

Recall -

Average Recall of the K-RE was computed as *89.61%.*

$R_{K\text{-}RE} = (\sum IKE_{K\text{-}RE\ Rel} / \sum IKE_{Total}) \times 100$

$R_{K\text{-}RE} = 89.61\%$.

Average Recall of the traditional manual requirement traceability approach was computed as 51.94 %.

$R_M = (\sum IKE_{Manual} / \sum IKE_{Total}) \times 100$

$R_M = 51.94\%$.

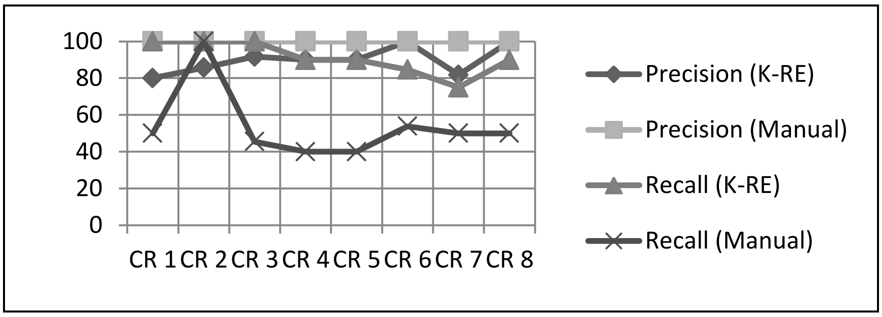

Fig. 7.3. Precision and Recall: Manual and K-RE based

7.6.1.6 Analysis

Average Precision of identifying knowledge elements impacted by the change requests when using K-RE was found to be 90.79%. If a knowledge element is being changed, K-RE uses the underlying domain knowledge ontology to identify all the related knowledge elements. For example, if it is required to change a ***feature***, the related ***rules***, complementary ***features***, ***use cases***, ***test cases*** will be highlighted as impacted elements. Only human intervention can discern if all need to be correspondingly updated. For example, not all ***test cases*** will need to be changed if a ***use case*** is being modified. As discussed earlier we have involved domain experts to review the results of manual identification and identification done by K-RE.

Average Recall of the K-RE was computed 89.61%. Recall using manual approach was found to be 51.94%. K-RE thus has an inherent ability to make knowledge visible using the domain knowledge ontology it incorporates. The business analysts could not identify as many impacted elements in the absence of such a mechanism. Some of the impacted elements such as ***tasks*** and ***validations*** associated with ***use cases***, complementary features were not obvious to them. K-

RE makes the impact easily visible to all stakeholders, and hence easy to discuss the effort involved.

7.6.2 Starting a new project using the Knowledge base

A ***Pension*** application for an insurance company in ***Europe Country2*** was to be developed. The KB created for ***Europe Country1*** was reviewed for reuse. From the repository, the knowledge elements mentioned in Table 7.7 were found to be closest to the new project for ***Europe Country2***. These were imported into the new project workspace from the repository.

Table 7.7. Knowledge elements reuse in project for Europe Country2

Knowledge elements in knowledge base from *Europe Country 1*	**Knowledge elements imported from *Europe Country 1* for reuse in *Europe Country 2* project**	**Remark(s)**	**Example**
52 Sub process	All of 52 State Pension sub processes and BPMN process maps.	18 out of the 52 imported State Pension sub processes were modified	Some of the modified sub processes are: *Contribution, Joiner, Risk Management, Customer Management*
220 Features	All of 220 features	Feature steps modified to accommodate country-specific variations.	Original Step: *Perform contribution limit check as dictated by Europe country 1* Modified step: *Perform contribution limit check as dictated by Europe country 2*
346 Business rules	250 business rules	50 new country-specific variants to business rules included.	New Business rule such as *In case of Europe Country 2, the member must hold a valid debit card were added*
1210 Business terms and relationships	All of 1210 business terms and relationships	Two hundred business entities and relations added afresh to the 1210 borrowed from repository.	Example: New terms such as *Growth rate, contribution* were added

Eighteen of the selected ***State Pension*** processes were modified. Two hundred ***business entities*** and relations were added afresh to the ones borrowed from repository. Fifty new country-specific variants to ***business rules*** had to be included. The resultant KB was reorganized, modified and refined in consultation with customers and domain experts, using the web 2.0 enabled communication in addition to the assistance that K-RE provides. For example, the analyst selected the ***process*** C*ontribution* to work with from the list of available process in the KB. She received a recommendation to add the ***processes*** *Customer management*, *Finance* and *Accounting*; as these processes are complimentary to the ***process*** *Contribution*. The analyst was also presented a list of features specific to selected process. For example, features such as *Member regular contribution by new direct debit, Member adhoc contribution by new direct debit, Member adhoc contribution by debit card* and *Member contribution correction* were presented to the analyst from the process *Contribution*. She selected the ***features*** - *member regular contribution by new direct debit* and *member adhoc contribution by new direct debit* to add to her project. K-RE prompted her to add ***feature*** - *Member contribution correction* as well (as feature - *member contribution correction* is complimentary to the other selected features). The Requirement analyst made changes to one of the features – *Member regular contribution by new direct debit*. She entered a new feature step – *'Verify member contribution details with respect to present growth rate and send notification to the scheme holder'*; K-RE provided an alert that *scheme holder* and *member* are synonymous terms, but *member* is the commonly accepted term. In addition to this, K-RE also parsed the text to identify new ***business terms*** and concepts (e.g. *growth rate*, *contribution*) and recommended the analyst to add them to the glossary. She was also prompted to add relevant ***business rules*** such as *'Growth rate must be within the range of 8% to 10%'*, ***validations*** such as *'Check if the growth rate is within the range of 8% to 10%'*. Corresponding ***use cases*** as well as ***screens*** were made visible to her to select from.

The final requirement specification for the new project consisted of 239 *State Pension* ***processes***, 3269 ***business entities*** and relations, 822 ***business rules*** along with exceptions and over-rider scenarios. A review by domain experts reveals that 60% of the knowledge needed to arrive at the baseline specification was reused from the repository created originally for the ***Europe Country1 Pension*** project. In subsequent phases 20-30% of time savings was observed.

The projects teams consider this to be a significant contribution to the requirements definition exercise; which otherwise starts from a clean slate for want of visible, accessible and configurable knowledge. The availability of a structured KB also serves as a 'thinking aid' for all RE stakeholders to brainstorm and arrive at a consensus. It is a easier to review and modify (if necessary) an existing feature or a process, than to come up with one afresh. Customers find it simpler to suggest changes and additions to an existing process or a rule than to narrate one from memory. Requirement analysts can leverage the domain vocabulary to have meaningful discussions with customers and optimize the time spent with SMEs.

7.7 Discussions and Conclusion

RE predominantly involves establishing a shared understanding of problem domain. It is estimated that more than 50% of requirements knowledge for similar projects can be reused completely or with minimal modification [2]. However, knowledge present in tacit form is not amenable to reuse. Even explicit knowledge in the form of disparate documents may not serve the purpose of reuse because it is not structured in a way that makes it visible, accessible and configurable. We address this need by developing a method and toolset to create, curate, and reuse knowledge for evolving requirements of large projects. K-RE facilitates extracting domain knowledge from semi structured and unstructured knowledge sources to create a structured KB consisting of generic requirement elements that can be reused in the same project as well as while starting new projects. We envisage a monitored environment where in a community of stakeholders create and evolve the generic KB to suit project specific needs. The just-in-time alerts to guide knowledge reuse in an RE process help achieve an improved completeness, consistency and richness of the resulting specification. We have used a combination of the social software principles and semantic web concepts.

We have demonstrated creation of knowledge repositories and its reuse in large projects.

In addition to knowledge reuse in RE, the concept of using active knowledge repositories can be extended to any exercise that draws on intensive knowledge services. We find that the foundation of our method and toolset is generic enough to cater to the knowledge reuse needs of stakeholders in very interesting emergent disciplines such as Nano technology based agriculture/food research [38, 39].

We realize however that the success of K-RE will depend largely on the quality of KB that we are able to create. Also, adopting K-RE would require a mindset change difficult to achieve in any organization. The upfront investment in creating a KB can also be a hindrance to adopting this approach.

7.8 Future work

The work presented here attempts to combine benefits of the inference and reasoning possibilities associated with the use of semantic web and the social aspects associated with web 2.0 for achieving knowledge reuse in RE.

Semantic web presents the additional possibility to link resources across the web by assigning persistent URIs to domain concepts and their relationships. We have recently attempted to explore this possibility for understanding and visualizing the multi-domain span of requirements in a given domain [40]. Our approach tested on a few sample user stories brings out that the approach can help in explicitly visualizing the multi-domain scope of requirements and improve their completeness at the stage of specification itself [40]. We are aware that the completeness of ontologies is a precursor to this method and hence a limiting factor to its successful application. We seek opportunities to further test the applicability and scalability of the method and tool in large projects.

Acknowledgments We thank the anonymous referees of our previously published work in MaRK 10 and MaRK 11 workshops at previous RE conferences. One of us (SG) would like to thank Vijaya Deepti, TCS for her solid support to this project and very many practical insights during the course of this work. The authors acknowledge inputs from Mini Jain and Preethu Rose (TCS) and reviews from V. S. Sivakumar and C. Sudheer of the Insurance unit of TCS.

References

[1] Kaiya H, Saeki M: Using Domain Ontology as Domain Knowledge for Requirements Elicitation. In: 14th IEEE International Requirements Engineering Conference, pp 189--198. IEEE Press, Minneapolis (2006)

[2] Lopez O, Laguna MA: Requirements Reuse for Software Development, in RE 01 Doctoral Workshop. 5th IEEE International Symposium on Requirements Engineering. Toronto, Canada, Aug 2001, pp 27--31. (2001)

[3] Feldman S, Sherman C: The high cost of not finding information, IDC Whitepaper, 2001.

[4] Making agile software development work for distributed teams, http://searchsoftwarequality.techtarget.com/news/article/0,289142,sid92_gci1277064,00.html. Accessed 15 Nov 2011

[5] Buchan J, Ekadharmawan CH, MacDonell SG: Insights into domain knowledge sharing in software development practice in SMEs. In: 16th Asia-Pacific Software Engineering Conference, pp 93--100. IEEE CS Press, Penang (2009)

[6] Lohmann S, Dietzold S, Heim P, Heino N: A web platform for Social Requirements Engineering, In: Munch, J., Liggesmeyer, P. (Eds), SENSE 09, Software Engineering in Social Software Environments, Series of the Gesselshaft fur Informatik (GI), 2009, Lecture Notes in Informatics (LNI), vol.150, pp 309--315. (2009)

[7] Ankolekar A, Krotzsch M, Tran T, Vrandecic D: The two cultures- mashing up web 2.0 and the semantic web, In: 16th international conference on World Wide Web, pp 825--834. ACM, Banff (2007)

[8] Maalej W, Happel H: A lightweight approach for knowledge sharing in distributed software teams, In: T. Yamaguchi (Ed.) PAKM 2008. LNAI 5345, pp 14--25, 2008, Springer-Verlag, Heidelberg (2008)

[9] IBM -Jazz, http://www-01.ibm.com/software/rational/jazz. Accessed 15 Nov 2011

[10] Ajmeri N, Sejpal R, Ghaisas S: A semantic and collaborative platform for agile requirements evolution, In: Third International Workshop on Managing Requirements Knowledge, pp 32--40. IEEE Press, Sydney (2010)

[11] Cheng B, Atlee J: Research directions in Requirements Engineering, In: Future of Software Engineering, pp 285--303. Minneapolis (2007)

[12] Flynn DJ: Information Systems requirements: determination and analysis, McGraw Hill London (1992)

[13] Hoffman L, Lehner F: Requirements Engineering as a success factor in software projects, In: IEEE software, vol. 18, pp 58--66. IEEE (2001)

[14] Boehm B: Software Engineering Economics, Upper Saddle River, NJ, Prentice Hall (1981)

[15] Kastanov A, Sakkinen M: Requirements quality control- A unifying framework, In: Requirements Engineering, vol 11, pp 42--57. Springer, New York (2006)

[16] Damian D, Chisan J: An empirical study of complex relationships between the requirements engineering process and other processes that lead to payoffs in productivity, quality and risk management, In: IEEE transactions in Software Engineering, vol 32, pp 433--453. IEEE, San Francisco (2006)

[17] Berners-Lee T, Hendler J, Lassila O: The Semantic Web. Scientific American, 5, 2001

[18] Shadbolt N, Berners-Lee T, Hall W: The Semantic Web Revisited, In: Intelligent Systems, vol. 21, pp 96--101. IEEE (2006)

[19] Hanneman A, Hocken C, Klamma R: Community driven Elicitation of Requirements with Entertaining Social Software, In: Munch, J., Liggesmeyer, P. (Eds), SENSE 09, Software

Engineering in Social Software Environments, Series of the Gesselshaft fur Informatik (GI), 2009, Lecture Notes in Informatics (LNI), vol.150, pp 317--328. (2009)
[20] Decker B, Ras E, Rech J, Jaubert P, Rieth M: Wiki based stakeholder participation in Requirements Engineering, In: IEEE Software, vol. 24(2), pp 28--35. IEEE CS Press (2007)
[21] Whitehead J: Collaboration in Software Engineering: A roadmap, In: Future of Software Engineering, pp 214--225. IEEE, Washington (2007)
[22] Folksonomy: http://vanderwal.net/folksonomy.html. Accessed 15 Nov 2011
[23] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS: Feature-oriented domain analysis (FODA) feasibility study. Tech. rep. CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University.
[24] Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification. Technical report, Object Management Group (OMG), February 2006.
[25] Cockburn A: Writing Effective Use Cases, Addison-Wesley, 2001.
[26] Ghazvinian A, Noy NF, Jonquet C, Shah N, Musen MA: What four million mappings can tell you about two hundred ontologies?, In: 8th International Semantic Web Conference, LNCS, vol. 5823, pp 229--242. Springer, Heidelberg (2009)
[27] McGuinness DL, Van Harmelen F (Eds.): OWL Web Ontology Language Overview, World Wide Web Consortium (W3C) recommendation, Feb. 2004 (2004)
[28] Ghaisas S: A method for identifying unobvious requirements in globally distributed software projects, In: Munch, J., Liggesmeyer, P. (Eds), SENSE 09, Software Engineering in Social Software Environments, Series of the Gesselshaft fur Informatik (GI), 2009, Lecture Notes in Informatics (LNI), vol.150, pp 297--308. (2009)
[29] Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M: SWRL- A semantic web rule language combining OWL and RuleML, W3C member submission. W3C, 21 May 2004 (2004)
[30] Kiu C, Lee CS: OntoDNA- Ontology Alignment Results for OAEI 2007, In: 6th International Semantic Web Conference (ISWC) and 2nd Asian Semantic Web Conference (ASWC), pp 196--205. (2007)
[31] Dao TN, Simpson T: Measuring Similarity between Sentences (online), http://opensvn.csie.org/WordNetDotNet/trunk/Projects/Thanh/Paper/WordNetDotNet_Semantic_Similarity.pdf. Accessed 15 Nov 2011
[32] Rusu D, Dali L, Fortuna B, Grobelnik M, Mladenic D: Triplet Extraction from Sentences, In: 10th International Multiconference, Information Society - IS 2007, Vol. A (2007), pp 218--222. (2007)
[33] MySQL: http://mysql.com. Accessed 15 Nov 2011
[34] OpenNLP Project: http://incubator.apache.org/opennlp. Accessed 15 Nov 2011
[35] WordNet : http://wordnet.princeton.edu. Accessed 15 Nov 2011
[36] Spijkerman W: Tool Support for Change Impact Analysis in Requirement Models- Exploiting semantics of requirement relations as traceability relations (2010)
[37] Bohner SA, Arnold RS: Software change impact analysis. IEEE CS Press, Los Alamitos (1996)
[38] Rose P, Bhat M, Vidhani K, Ajmeri N, Gole A, Ghaisas S: Intelligent informatics platform for nano-agriculture, In: 11th IEEE Conference on Nanotechnology (IEEE-NANO), 2011, pp 916--919. IEEE Press (2011)
[39] Rose P, Gole A, Ghaisas S: A semantic regulatory framework for nanotechnology application in agri-food domain, In: Fourth International Workshop on Requirements Engineering and Law (RELAW), 2011, pp 60--66. IEEE Press (2011)
[40] Ajmeri N, Vidhani K, Bhat M, Ghaisas S: An Ontology-based Method and Tool for Cross-Domain Requirements Visualization, In: Fourth Workshop on Managing Requirements Knowledge, MaRK11, pp 22--23, IEEE Press (2011)