A KNOWLEDGE ENGINEERING APPROACH SUPPORTING COLLABORATIVE WORKING ENVIRONMENTS BASED ON SEMANTIC SERVICES

Celson Lima, Paulo Figueiras and Ruben Costa
UNINOVA, Centre of Technology and Systems, Campus da Caparica, Quinta da Torre
2829-516 Monte Caparica, Portugal

Keywords: Collaboration, Knowledge Management, Semantic Services, Semantic Reasoning, Ontology.

Abstract: This paper brings a contribution focused on collaborative engineering projects where knowledge plays a key role in the process. Collaboration is the arena, engineering projects are the target, knowledge is the currency used to provide harmony into the arena since it can potentially support innovation and, hence, a successful collaboration. Innovation often happens when knowledge (existing, recycled, or new) is combined and it depends on individuals (or groups) holding the appropriate knowledge to provide the required breakthrough. This work aims to support collaborative work carried out by project teams, through a set of knowledge-enabled services context aware. We introduce our conceptual approach (and its respective implementation) supporting a modular set of semantic services based on individual collaboration in a project-based environment, the CoSpaces Knowledge Support (CoSKS) component. CoSKS provides semantic based classification, reasoning and context analysis processes, to support the instantiation of the knowledge spiral and transform it into a semantically contextualized knowledge tree, made out of concepts that best represent contexts. Results achieved so far and future goals pursued by this work are also presented here. This work has been conducted as part of the CoSpaces Integrated project, funded by the European Commission.

1 INTRODUCTION

Over the last two decades, the adoption of the Internet as the primary communication channel for business purposes brought new requirements especially considering the collaboration centred on engineering projects. By their very nature, such projects normally demand a good level of innovation since they tackle highly complex challenges and issues. On one hand, innovation often recurs to combination of knowledge (existing, recycled, or brand new) and, on the other hand, it depends on individuals (or groups) holding the appropriate knowledge to provide the required breakthrough.

Engineering companies are project oriented and successful projects are their way to keep market share as well as to conquer new ones. Engineering projects strongly rely on innovative factors (processes and ideas) in order to be successful. From the organisation point of view, knowledge goes through a spiral cycle, as presented by Nonaka and Takeuchi (Nonaka & Takeuchi, 1995). It is created and nurtured in a continuous flow of conversion, sharing, combination, and dissemination, where all the aspects and contexts of a given organisation, are considered, such as individuals, communities, and projects.

Knowledge is considered the key asset of modern organisations and, as such, industry and academia have been working to provide the appropriate support to leverage on this asset (Firestone & McElroy 2003). Few examples of this work are: the extensive work on knowledge models and knowledge management tools, the rise of the so-called knowledge engineering area, the myriad of projects around ‘controlled vocabularies’ (i.e., ontology, taxonomies, etc..), and the academic offer of knowledge-centred courses (graduation, master, doctoral) .

The quest for innovation to be used a wild card for economic development, growing and competitiveness, affects not only organisations, but also many countries. This demand for innovative processes and ideas, and the consequent pursuit of

Lima C., Figueiras P. and Costa R..
A KNOWLEDGE ENGINEERING APPROACH SUPPORTING COLLABORATIVE WORKING ENVIRONMENTS BASED ON SEMANTIC SERVICES.
DOI: 10.5220/0003102801230132
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2010), pages 123-132
ISBN: 978-989-8425-29-4

effectively more knowledge, raise inevitably issues regarding the adoption and use of Knowledge Management (KM) models and tools within organisations.

As relevant literature shows (Koening 2002; Malhotra 1999; McElroy 1999; Dalkir 2005), KM does not only comprise creation, sharing, and acquisition of knowledge, but also classification, indexation, and retrieval mechanisms. Knowledge may be classified by its semantic relevance and context within a given *environment* (i.e., the organisation itself or a collaborative workspace). This is particularly useful to: (i) improve collaboration between different parties at different stages of a given project life cycle; and (ii) to assure that relevant knowledge is properly capitalised in similar situations. For example, similar projects can be conducted in a continuously improved way if lessons learned from previous are promptly known when a new (and similar to some previous one) project is about to begin.

The CoSKS is a software component of a collaborative engineering environment being developed to support real-time collaboration, providing project teams with ontology-enabled services and proactive capabilities, targeting the improvement of agility and semantic richness in the decision making process, during the execution of a engineering project. CoSKS conceptually covers three major dimensions, namely collaboration, knowledge and reasoning (Costa et al., 2010). Collaboration targets behavioural aspects (e.g. pro-activity, reactivity, autonomy, etc.) and achievement of shared goals (Costa et al., 2010). Knowledge, the dimension particularly explored in this paper, relates to the 'currency' being exchanged during a collaborative process, in this case a collaborative engineering process. Technical documents, lessons learned, expertises, etc., are some examples of such currency. Reasoning relates to the use of data and text mining techniques to support the knowledge life cycle during a given collaborative process.

This paper is structured as follows: Section 2 defines the problem to be tackled. Section 3 covers the state of practice related to this work. Section 4 introduces the software components handling the knowledge related matters previously introduced. Section 5 gives illustrative examples of the software operation. Finally section 6 concludes the paper and points out the future work to be carried out.

2 PROBLEM DEFINITION

The research problem driving this work is two-folded: (i) *which model and tools could be developed in order to make the current collaborative decision making process on engineering projects more agile*?; and (ii) *what could be both conceptual and technical foundations to be adopted and adapted in order to develop such tools*? Our hypothesis is that *"agility" on the decision making process on collaborative engineering projects can be achieved if knowledge elements are used as the 'currency' to enhance collaborative interactions supported by reasoning mechanisms.* Knowledge elements shall be contextualised by self-adaptive semantic components which can be reused using reasoning mechanisms in order to match problems and solutions.

The approach followed here is centred on a problem-solution representation, enabling users to keep track of problems occurred and decisions made to solve them which can be reused whenever necessary to solve new problems. The technical development supporting this work relies into tree distinct dimensions, namely: (i) a *behavioural* capability which complement the human ability to act on a context of uncompleted information; (ii) a *reasoning* mechanism able analyze and extract conclusions from pre-existent knowledge; and (iii) *semantic* services in order to provide meaning under the context of each application scenario environment, decision making, and semantic. They are implemented through the following elements: Computer-Supported Cooperative Work (CSCW) infrastructure, a set of ontology-enabled services, and (data and text) mining services. It is worth noticing that this is an ongoing research under validation and, as such, results presented here are preliminary ones.

Figure 1 depicts the three main dimensions which support the instantiation of a collaborative engineering project environment and provide the foundations of this work.

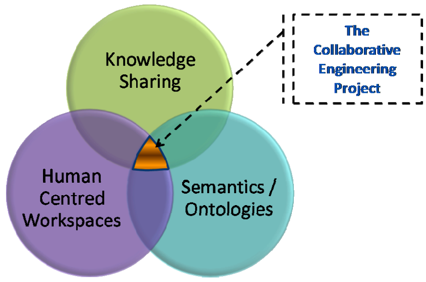

Figure 1: The Collaborative Engineering Project.

As previously presented, innovation may arise through the capitalisation of knowledge (already existing or new one) hold by individuals or groups. Nonaka and Takeuchi (1995), argued that knowledge goes through an evolving spiral when it is transformed from tacit (the inner knowledge, intangible) to explicit (visible, the tangible one) knowledge. They represent this process through the SECI model, which covers the four transformation processes involving the two knowledge types, namely: Socialisation (from tacit to tacit), Externalisation (from tacit to explicit), Combination (from explicit to explicit), and Internalisation (from explicit to implicit).

The success of collaboration considering an engineering project, where project teams are working together targeting a shared goal, essentially relies on capitalising on the existing knowledge as well as being capable to find innovative solutions to faced problems. Therefore, we can see the instantiation of the SECI model within the collaborative engineering environment towards agile decision making process, where knowledge is: (i) transformed in a evolving way along the time; (ii) managed around problems and solutions in order to be proper capitalised (Costa et al., 2010); (iii) better capitalised with the appropriate support of reasoning mechanisms; and (iv) supported by a set of ontology-enabled services to increase semantics.

Knowledge needs to be shared in order to be proper capitalised during decision making processes. On one hand knowledge sharing is heavily dependent on technical capabilities and, on the other hand, since the social dimension is very strong during collaboration, there is also an increased need to take into account how to support the culture and practice of knowledge sharing. For instance, issues of trust are critical in collaborative engineering projects, since the distribution of knowledge and expertise means that it becomes increasingly difficult to understand the context in which the knowledge was created, to identify who knows something about the issue at hand, and so forth.

3 THE COSKS FRAMEWORK

3.1 Basic Concepts

The key concepts supporting this work, described in this section, are the following: *Decisional Gates*, *Knowledge Elements & Semantics*, and *Context*.

Projects are conducted through a series of meetings and every meeting is considered a Decisional Gate (DG), a convergence point where decisions are made, problems are raised, solutions are (likely) found, and tasks are assigned to project participants. Pre-existing knowledge serves as input to the DG, the project is judged against a set of criteria, and the outputs include a decision (go/kill/hold/recycle) and a path forward (schedule, tasks, to-do list, and deliverables for next DG). (figure 2).

Each DG is prepared (through the creation of agendas), and the events that occur during the meeting shall be recorded. Between two DGs there is a permanent monitoring on the execution of all tasks executed. After meeting closure, there is a need for a mechanism to enable the preparation the minutes easily, highlighting the major decisions that were made during the meeting.

DGs normally go through the following phases: (i) Individual work; (ii) Initialisation; (iii) Collaboration; and (iv) Closing/Clean-up. Individual work relates to asynchronous collaboration, where all individuals involved in the project are supposed to provide inputs to the undergoing tasks. *Initialisation* (pre-meeting) covers the preparation of the meeting agenda and the selection of the meeting participants. *Collaboration* phase is the meeting itself where participants try to reach a common understanding regarding the issues from the agenda, using the right resources. This phase also considers the annotation of the decisions made during the meeting. Finally, *Closing/Clean-up* basically targets the creation of meeting minutes.

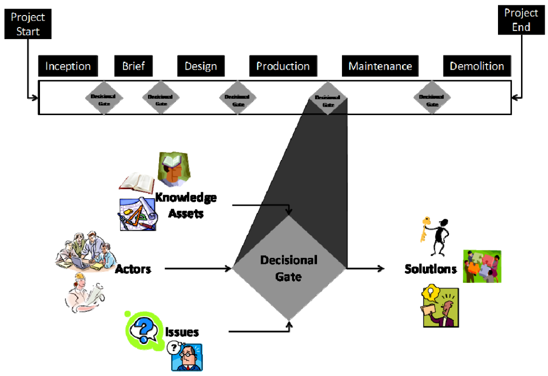

Figure 2: The Decisional Gate.

Other basic definition adopted here is *Knowledge Element* (*K-Elem*). It represents pieces of knowledge that can be captured, stored, published, shared, and reused among the project teams. K-Elem is the relevant knowledge to provide the proper support to e-collaboration in a given project. Users will reason in terms of K-Elems. The system has been conceived and essentially works around the K-Elems. In addition to ordinary documents, some specific examples of K-Elems used are: project, issues, solutions, agendas, minutes, tasks, participants, and project post-mortem (figure 3).

K-Elems strongly rely on ontological concepts, as a way to reinforce their semantic links. The CoSKS ontology uses a taxonomy of concepts holding two dimensions: on one hand, the knowledge elements themselves are represented in a tree of concepts and, on the other hand, the industrial domain being considered (in this case, the Construction industry). Instances of concepts (also called individuals) are used to extend the semantic range of a given concept. For instance, the ontological concept of 'Design_Actor' has two instances to represent architect and engineer as roles that can be considered when dealing with K-Elem related to design (experts, design-related issues/solutions, etc.). Moreover, each ontological concept also includes a list of terms and expressions, called *equivalent* terms, which may represent synonyms or expressions that can lead to that concept. Ontology support is particularly useful in terms of indexation and classification towards future search, share and reuse.

Figure 3: The Knowledge Elements.

The CoSKS ontology is developed to support and manage the use of expressions which contextualize a K-Elem within the knowledge repository. The ontology adds a semantic weight to relations among K-Elems stored into the knowledge repository. Every ontological concept has a list of 'equivalent terms' that can be used to semantically represent such concept. These terms are, then, treated in both statistical and semantic way to create the semantic vector that properly indexes a given K-Elem.

The CoSKS ontology was not developed from scratch; rather, it has been developed taking into account relevant sources of inspiration, such as the buildingsmart IFD model (BuildingSmart 2010), omniclass (omniclass 2010), and the e-cognos project (Lima et. al 2002).

Finally, the definition of Context is required. It is easily understood that *experts* (from different areas of expertise) working collaboratively in a given product have different needs/visions about/on the knowledge used, which is strongly influenced by their backgrounds, roles, responsibilities, etc.. Additionally, different *types of projects* can give different uses to the same knowledge (e.g. knowledge about accessibility regulations used in public versus private project buildings). Going further, knowledge can be treated differently depending on the *meeting* it is captured/used (e.g. deviations and delays in different *phases* of the project have highly different meanings). In different tasks the same knowledge may have different uses. The issue to be solved also defines the relevance of a given knowledge. All terms written in italic compose the preliminary list of valid contexts adopted here.

3.2 The CoSKS Technical Foundations

The CoSKS technical framework is structured into four layers (figure 4), namely: Presentation, Behavioral, Service, and Knowledge.

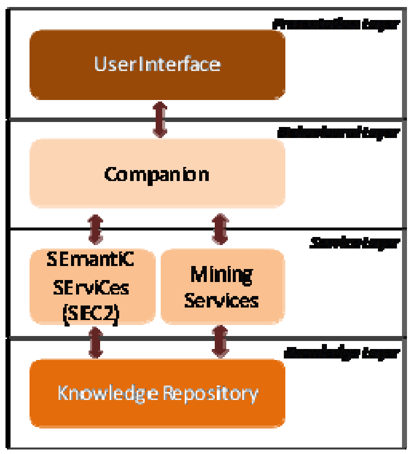

Figure 4: The CoSKS Layers.

The Presentation layer supports the interaction with the CoSKS user, through a web portal, which represents the collaboration workspace environment where users exchange and use pre-existent knowledge. The Companion component implements both proactive and reactive behaviours of CoSKS.

The *Mining Services* provide CoSKS with reasoning-related capabilities, which are used to discover useful knowledge aiming to identify patterns of problems and solutions and establish the relationships between them. These services are used as a way to anticipate problems and find potential solutions. The main capabilities provided are:

The *SEmantiC SErviCes* (SEC2) are the central focus of this paper, which provide semantic capabilities in order to support the CoSKS operation. It acts as a middleware between knowledge elements and behavioral layers, offering the following functionalities: semantic contextualisation and filtering for K-Elems, creation of semantic vectors, semantic vector based indexation and retrieval of K-Elems.

3.3 Contextualisation Process using Semantic Vectors

In order to provide agility on the decision making processes of collaborative engineering projects, the semantic services offered by CoSKS essentially depend on the contextualisation of K-Elems.

The basis for context definition lies on implementation of semantic vectors. Each semantic vector contains the necessary ontological concepts that best represent a given K-Elem when it is stored into the knowledge repository. These concepts are ordered by their semantic relevance regarding the K-Elem acquisition context. K-Elems are compared and matched based on their semantic vectors and the degree of resemblance between semantic vectors directly represents the similarity between K-Elems contexts. To better understand the CoSKS contextualisation process through semantic vectors comparison (Figure 5), it is necessary to understand how and where these are created and used.

Semantic vectors are automatically created using project-related knowledge, gathered from the knowledge repository, using data and text-mining techniques. The mining process collects words and expressions, to be matched against the equivalent terms which represent the ontological concepts. This produces an inventory of: (i) the number of equivalent terms matched at each ontological concept; and (ii) the total number of equivalent terms necessary to represent the harvested knowledge. This inventory provides the statistical percentage of equivalent terms belonging to each ontological concept represented in the universe of harvested knowledge. This step represents, the calculus of the 'absolute' semantic vector of a given K-Elem, taking into account the equivalent terms-based percentages.

However, the approach presented here also considers a configurable hierarchy of K-Elems' relevance, as part of the creation of semantic vectors. This hierarchy is defined using 'relative' semantic factors to all types of K-Elems, which ranges respectively from low relevance (0) to high relevance (1) for the context creation. Both hierarchy and relative semantic factors are originally proposed by SEC2, but they can be changed if necessary, depending on what K-Elems are considered most relevant for the contextualization process. For illustrative purposes only, an example of this hierarchy could be: issues (1), solutions (1), experts (0.7), Post-mortem (0.7), etc..

The final step, which comprehends the semantic evaluation, also includes ontological concepts that are not linked to the knowledge gathered, but have a semantic relationship of proximity with a relevant (heavy) ontological concept. This is done through the definition of a secondary semantic factor to ontological concepts based on their relative distances, inside the ontology tree.

Summing up, the final calculation of the semantic vector includes: statistical percentages based on the equivalent terms, the hierarchy of relevance for K-Elems, and the weight assigned to the proximity level.

As referred above, semantic vectors are continuously updated through the project's life cycle, and even in project's post-mortem. This is done in order to maintain the semantic vector's coherence with the level of knowledge available. Semantic vectors are automatically created: (i) whenever a new K-Elem is gathered; and (ii) to help answering queries issued by the users.

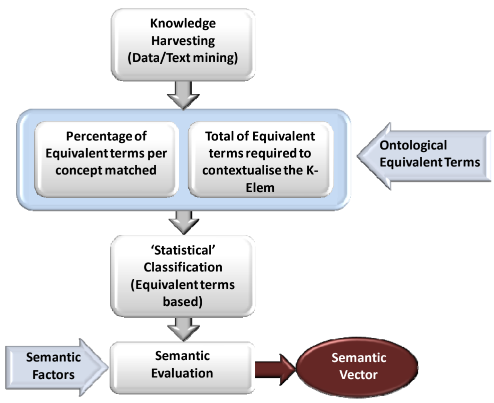

Figure 5: Creation of Semantic Vectors.

Two types of queries are supported by SEC2. The first type corresponds to context-based queries relative to projects' issues. These queries are used to help finding solutions to those issues, capitalizing on existing K-Elems, which can come from similar projects for instance, in the form of issues with similar contexts and their respective solutions, tasks, documents and experts involved, etc..

The second type of query is based on free text search. When a free text query is issued, it is processed taking into account the user's semantic context. This is made through the dynamic definition (by the user) of 'relative' semantic factors. As previously described, these factors have an impact on the calculation of the semantic vector reflecting, in this case, the query itself. Hence, the query is transformed into a semantic vector, through semantic indexation of the query text with the respective factors.

4 THE SEC2 COMPONENT

Recalling fact that this work targets the Construction industry, the domain ontology was essentially built following guidelines from the international references of this sector, namely the Omniclass Construction Classification System (OCCS, 2010), the e-COGNOS project (Lima et al., 2002) and BuildingSmart IFD (BuildingSmart 2010).

Broadly speaking, OCCS is composed by a collection of tables which represent the concept families that define construction projects in their different perspectives. As previously described, the CoSKS ontology provide semantic values to words and expressions which denote a semantic relation, directly (synonyms) or indirectly (semantic related expressions), with the main concepts that characterize the context of a construction project.

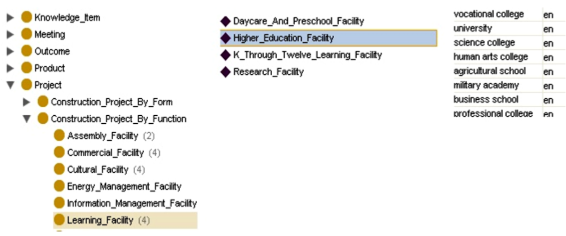

Figure 6: Excerpt II of SEC2 Ontology.

Figure 6 illustrates the use of equivalent terms, using as example the *Higher_Education_Facility* individual, which is a sub-entity of the *Learning_Facility* concept. In this example, equivalent terms are associated to the *Higher_Education_Facility*, such as "university" or "business school". Equivalent terms extend the semantic range of the individual they are related to. It is worth emphasizing that the equivalent terms were/are obtained from the last hierarchy levels of OCCS or from technical controlled vocabularies (Lima, Zarli and Storer 2007) used in Construction.

In this sense, the ontology can be described as a resource that can semantically represent several contexts found in collaborative engineering projects in the Construction sector. Additionally, it has been developed using the W3C recommended OWL (OWL, 2004) language using an ontology editor tool (Protégé, 2010).

From a technical perspective, SEC2 is an application conceived to be an open and flexible middleware, in a sense that it can handle other ontologies from different knowledge areas or industrial sectors, as long as they are represented in OWL and follow the structure proposed here.

SEC2 includes an API (Jena, 2010), which is used to build a persistent ontological model in the CoSKS knowledge repository. This model is represented in relational database, enabling online ontology storage and update, through functions provided by API.

The SEC2 K-Elem Repository stores all K-Elems currently available into the system, together with their respective semantic vectors. The following K-Elems are stored: Project, Organisation, Issue, Solution, Task, Meeting, Minute, Agenda, Actor, and Role (actor type).

5 EXAMPLES

For illustrative purposes, this section describes examples of context indexation of K-Elems as well as a free text search query.

5.1 K-Elem Context Indexation

Consider, for instance, that a new issue is registered into the SEC2 component, and such issue is related to a project with the following specifications (also stored into the database).

- Title: Building project for an University, Lisbon, Portugal.
- Description: The project is based on the construction of a university building near Lisbon, constituted by fifty class rooms and twenty laboratories, in a mid-rise fashion. The

building has X meters in height and a total area of Y square meters, and has a minimal building design, located on a slight slope with 3% of slope rating.

- Start Date: March, 10th 2010.
- End Date: December, 20th 2011.
- Real Days: 30 days.
- Project Phase: Design.
- Project Type: University mid-rise building.

The project has some intervening actors on the project itself and on the tasks that will be part of the solution to the issue in question.

Consider an architect actor that has been allocated to the task associated to the issue's solution, or that has some relevance in the issue context. The database entry for this actor type, in the project's domain, is something like:

- Actor Type: Architect;

The issue is also described in the database entry and it contains data of vital importance to the semantic categorisation of the issue itself:

- Issue Title: Building design plan measures issue;
- Description: The building design plan has a measurement error, generated by the misplacement of a column or pillar in the drawing;
- Problems: The column is misplaced by Z centimeters on the east faced, creating a misplacement of both the wall and the column; The space between the wall and the pillar is not correct; The column is made out of steel, and the wall is a normal cement wall with steel foundations;
- Solutions: No solutions yet;
- Deviations: No deviations yet;

In addition to the knowledge extracted from K-Elem like *Project*, *Actor*, and *Issue*, there is still much more issue-related information on the SEC2 repository, namely knowledge related to *Task* and *Task_Actors*, as well as all documents and respective metadata related to both project and issue. Now consider a task, allocated to the architect actor described above, with the following specifications:

- Title: Building plan redrawing task;
- Description: The building plan needs a redrawing correction, The correction can be made in two ways: Erasing the column or redrawing the column in a new location.
- Problems: No problems yet.
- Solutions: No solutions yet.
- Deviations: No deviations yet.

Considering that information presented above is present on CoSKS, the first step is to gather the expressions context-related to the issue, through data mining techniques.

Expressions which seem to have a higher semantic relevance are: "University", "Mid-rise building", "Building design", "Design phase", "Architect", "Measures issue", "Measurement error", "Misplacement", "Column", "Pillar", "Drawing", "Wall", "Steel", "Cement", "Steel foundations", "Building plan" and "Redrawing task".

Presented in this manner, the information gathered is represented in a disperse set, without any semantic added-value. However, it is still possible to understand that the most relevant concepts to contextualize an issue, ordered by relevance, are:

- The problem itself and its associated tasks, since they contain information related to the kernel of the issue;
- The professional involved with: in this case an architect since this issue is purely architectonic; and
- The project type and function because there are also structural aspects to be taken into account.

The problem appears when the issue's contextualisation process is formalized by a software tool, and not a human brain, i.e. the contextualization process is achieved by means of the usage of text mining algorithms which automatically extract relevant expressions from non structured information. Hence, the second step is to semantically enrich the gathered expressions, allowing them to be processed and classified. This semantic value is achieved through the comparison of the gathered expressions against the ontological concepts. In this example, the result from this comparison is presented in the following format: "equivalent term"; Individual; Class; ABSOLUTE PARENT CLASS:

- "University"; Higher_Education_Facility; Learning_Facility; PROJECT BY FUNCTION
- "Mid-rise building"; Mid_rise_Building; PROJECT_BY_FORM
- "Building design"; Architect; Design_Actor; ACTOR
- "Design phase"; Design_Phase; PROJECT_BY_PHASE
- "Architect"; Architect; Design_Actor; ACTOR
- "Measures issue"; Measures_Issue; Technical_Issue; ISSUE
- "Measurement error"; Measures _Issue; Technical_Issue; ISSUE
- "Column"; Structural_Frame; Structural_And_Space_Division_Product; PRODUCT

- "Pillar"; Structural_Frame; Structural_And_Space_Division_Product; PRODUCT
- "Drawing"; Architect; ACTOR & Drawing; KNOWLEDGE_ITEM
- "Wall"; Structural_Wall; Structural_And_Space_Division_Product; PRODUCT
- "Cement"; Binding_Agent; General_Purpose_Construction_Accessory_And Surfacing_Product; PRODUCT
- "Building plan"; Architect; Design_Actor; ACTOR
- "Redrawing task"; Redrawing_Task; Technical_Task; TASK

After matching those, the next step is to gather equivalent terms matched for each ontological concept, asserting the total number of equivalent terms matched and the number of equivalent terms corresponding to each ontological concept:

- Structural Frame: "Column"; "Pillar" (2)
- Structural Wall: "Wall" (1)
- Binding Agent: "Cement" (1)
- Architect: "Architect"; "Building design"; "Drawing" (3)
- Redrawing Task: "Redrawing task" (1)
- Measurement Issue: "Measures issue"; "Measurement error" (2)
- Mid-Rise Building: "Mid-rise building" (1)
- Higher Education Facility: "University" (1)
- Design Phase: "Design phase" (1)
- Drawing: "Drawing" (1)

The total of equivalent terms matched is fourteen (14). Even though gathered knowledge is now quantified and semantically organized, it does not provide the issue's contextualisation. The next step is, then, to calculate the percentages of equivalent terms matched for each K-Elem, through statistic calculus, using the formula:

$$\%_{Ontological\ Concept} = \frac{n}{N} \times 100 \tag{1}$$

where n is the number of equivalent terms matched for each K-Elem, and N is the total number of equivalent terms matched. Hence:

$$\%_{Structural\_Frame} = \frac{2}{14} \times 100 = 14{,}3\% \tag{2}$$

$$\%_{Structural\_Wall} = \frac{1}{14} \times 100 = 7{,}4\% \tag{3}$$

$$\%_{Binding\_Agent} = \frac{1}{14} \times 100 = 7{,}4\% \tag{4}$$

$$\%_{Architect} = \frac{3}{14} \times 100 = 21{,}4\% \tag{5}$$

$$\%_{Redrawing\_Task} = \frac{1}{14} \times 100 = 7{,}4\% \tag{6}$$

$$\%_{Measurement\_Issue} = \frac{2}{14} \times 100 = 14{,}3\% \tag{7}$$

$$\%_{Design\_Phase} = \frac{1}{14} \times 100 = 7{,}4\% \tag{8}$$

$$\%_{Mid-Rise\_Building} = \frac{1}{14} \times 100 = 7{,}4\% \tag{9}$$

$$\%_{Higher\_Education\_Fac.} = \frac{1}{14} \times 100 = 7{,}4\% \tag{10}$$

$$\%_{Drawing} = \frac{1}{14} \times 100 = 7{,}4\% \tag{11}$$

As one can see, even though results are semantically classified through ontological equivalent terms, compared and statistically transformed into percentages, they do not define the accurate context of the given issue.

The next process applied on gathered, classified, and calculated knowledge, provides a semantic factor hierarchy to the calculated results, by attributing factors of importance to each ontological concept with matched equivalent terms.

As referred before, knowledge associated to the issue itself and respective tasks should possess higher semantic relevance in the contextualisation, followed by the actor, the project, etc.. However, statistic results still do not reflect the previous inference. Therefore, the attributed semantic factors are:

- Issue: 30%.
- Task: 20%.
- Actor: 15%.
- Project Phase: 10%.
- Project Form: 10%.
- Project Function:7%.
- Product:5%.
- Knowledge Item: 3%.

Semantic factors are applied using the following formula:

$$sw_{concept} = \%_{concept} \times F_{concept} \tag{12}$$

$sw^{*}_{Concept}$ is the first form of semantic weight associated to a given ontological concept, $\%_{Concept}$ represents the relevance percentage of each ontological concept, and $F_{Concept}$ is the semantic factor applied to the such a concept. Hence:

$$sw_{Measurement\_Issue} = 0{,}143 \times 0{,}3 == 0{,}0429 \quad (13)$$

$$sw_{Redrawing\_Task} = 0{,}074 \times 0{,}2 == 0{,}0148 \quad (14)$$

$$sw_{Architect} = 0{,}214 \times 0{,}15 = 0{,}0321 \quad (15)$$

$$sw_{Design\_Phase} = 0{,}074 \times 0{,}1 = 0{,}0074 \quad (16)$$

$$sw_{Mid-Rise\_Building} = 0{,}074 \times 0{,}1 == 0{,}0074 \quad (17)$$

$$sw_{Higher\_Education\_Fac.} = 0{,}074 \times 0{,}07 = = 0{,}0005 \quad (18)$$

$$sw_{Structural\_Frame} = 0{,}143 \times 0{,}05 == 0{,}0072 \quad (19)$$

$$sw_{Structural\_Wall} = 0{,}074 \times 0{,}05 == 0{,}0037 \quad (20)$$

$$sw_{Binding\_Agent} = 0{,}074 \times 0{,}05 = 0{,}0037 \quad (21)$$

$$sw_{Drawing} = 0{,}074 \times 0{,}03 = 0{,}0022 \quad (22)$$

It is easy to see that these semantic weights are not heavy. In order to solve this result incoherence, another statistic procedure is applied. First, all the above results are summed, and then a percentage is applied to produce the new semantic weight of each ontological concept using the result of the sum, according to the following expression:

$$SW_{Ontological\ Concept} = \frac{sw_{Ontological\ Concept}}{\sum sw_i} \times 100 \quad (23)$$

where $SW_{Ontological\ Concept}$ represents the final semantic weight of a given ontological concept, and $\sum sw_i$ is the total sum of all the first forms of semantic weights, which is:

$$\sum sw_i = 0{,}1219 \quad (24)$$

Therefore:

$$SW_{Measurement\_Issue} = 35{,}2\% \quad (25)$$

$$SW_{Architect} = 26{,}3\% \quad (26)$$

$$SW_{Redrawing\_Task} = 12{,}1\% \quad (27)$$

$$SW_{Design\_Phase} = 6{,}1\% \quad (28)$$

$$SW_{Mid-Rise\_Building} = 6{,}1\% \quad (29)$$

$$SW_{Structural\_Frame} = 5{,}9\% \quad (30)$$

$$SW_{Structural\_Wall} = 3{,}0\% \quad (31)$$

$$SW_{Binding\_Agent} = 3{,}0\% \quad (32)$$

$$SW_{Drawing} = 1{,}9\% \quad (33)$$

$$SW_{Higher\_Education\_Fac.} = 0{,}4\% \quad (34)$$

The semantic weights presented above define the semantic vector of the issue used here. The final step is a comparison between the created semantic vector and the semantic vectors of other issues. These are classified through their structural resemblance with the former one.

5.2 The Free Text Search

Consider the scenario where the architect assigned to a task concerning the issue created on the previous example, performs a free text search, in order to find another architect which has already worked on a similar issue, who could be knowledgeable on technical design and have decision making skills.

The free text query could be issued as follows: "architect, skilled in technical design and decision making, and that has been working for a redrawing task, associated to a measurement error". As in the previous example, comparison, statistic and semantic processes are applied to the query. The first step is to extract relevant knowledge from the query text, in the form of regular expressions and words.

In this case, the extracted expressions would be: "architect", "technical design", "decision making", "redrawing task" and "measurement error".

The next step is to classify the extracted knowledge, matching it with ontological keywords ("equivalent terms"; Individual; Class; ABSOLUTE PARENT CLASS):

- "Architect"; Architect; Design Actor; ACTOR
- "Technical Design"; Technical Design; Technical Skill; SKILL
- "Decision Making"; Judgement And Decision Making; Systems Skill; SKILL
- "Redrawing task"; Technical Task; TASK
- "Measurement error"; Measures Issue; Technical Issue; ISSUE

Thus, using equation (1), with N equal to 5:

$$\%_{Architect} = \frac{1}{5} \times 100 = 20{,}0\% \quad (35)$$

$$\%_{Technical\_Design} = \frac{1}{5} \times 100 = 20{,}0\% \quad (36)$$

$$\%_{Judgement\_And\_Decisoin\_Making} = \frac{1}{5} \times 100 = 20{,}0\% \quad (37)$$

$$\%_{Technical\_Task} = \frac{1}{5} \times 100 = 20{,}0\% \quad (38)$$

$$\%_{Redrawing\_Task} = \frac{1}{5} \times 100 = 20{,}0\% \qquad (39)$$

6 CONCLUSIONS

This paper brings a contribution focused on collaborative engineering projects where knowledge engineering plays the central role in the decision making process.

Key focus of the paper is the SEC2 component, which essentially provides semantic services enabled by a domain ontology. This work specifically addresses collaborative engineering projects from the Construction industry, adopting a conceptual approach supported by knowledge-based services and reasoning mechanisms. The knowledge elements contextualization process is supported using a semantic vector holding a classification based on ontological concepts. Illustrative examples showing the process are part of this paper.

When addressing collaborative working environments, there is a need to adopt a semantic description of the preferences of the users and the relevant knowledge elements (tasks, documents, roles, etc..). In this context, we foresee that Ontologies which support semantic compatibility for specific domains should be self-adaptive and self-evolving within a particular context.

The same way that knowledge by itself is an evolving process, ontologies should also be resilient whenever new knowledge is generated and new concepts are created. Ontologies ability to adapt to different environments and different context of collaboration is of extremely importance, when addressing collaborative engineering projects at the organizational level. Resilient ontologies is a topic which implies deeper research within the scope of this work.

REFERENCES

Buildingsmart IFD Accessed 2010-05-20 at http://www.ifd-library.org/index.php/Main_Page

Cao, L. (2008). Behavior Informatics and Analytics: Let Behavior Talk. IEEE International Conference on Data Mining Workshops, (pp. 87-96). Sydney, Australia.

Costa, R., Lima, C., Antunes, J., Figueiras, P. and Parada, V. (2010), Knowledge Management Capabilities Supporting Collaborative Working Environments in a Project Oriented Context. In 2nd *ECIC, Lisbon. European Conference on Intellectual Capital*, pp. 208-216.

Vorakulpipat, C., Rezgui, Y. (2008). An evolutionary and interpretive perspective to knowledge management, *Journal of Knowledge Management*. Vol. 12 No. 3, pp. 17-34.

Cheverst, K., Byun, H. E., Fitton, D., Sas, C., Kray, C., & Villar, N. (2005). Exploring Issues of User Model Transparency and Proactive Behaviour in an Office Environment Control System. *User Modeling and User-Adapted Interaction* , pp. 1-39

CoSpaces, Innovative Collaborative Work Environments for Design and Engineering. Accessed 2010-03-20 at www.cospaces.org

Dalkir, M. (2005). "Knowledge Management in Theory and Practice". Elsevier

Davenport, T. and Prusak, L. (1998), Working Knowledge: How Organizations Manage What They Know, *Harvard Business School Press, Boston*, MA.

Firestone, J. M. and McElroy, M. W. (2003), Key Issues in the New Knowledge Management, Butterworth-Heinemann, Woburn, MA.

Jena, Accessed 2010-05-20 at jena.sourceforge.net/

Koenig, M. E. D. (2002), ''The third stage of KM emerges'', KMWorld, Vol. 11 No. 3, pp. 20-1.

Liang, D. 1998. Virtual Organisations, a New Organisational Model for the Millennium. Accessed Sept 22, 2009: www.smuts.uct.ac.za/~dliang/essays.

Lima, C., Zarli, A., Storer, G. (2007), Controlled Vocabularies in the European Construction Sector: Evolution, Current Developments, and Future Trends. *In the Proceedings of CE2007, 14th ISPE International Conference in Concurrent Engineering*, July 2007, São José dos Campos, Brazil, pp. 565-574, ISBN 978-1-84628-975-0.

Lima, C., Fies, B., El Diraby, T., Lefrancois, G. (2003), The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience . In: 10TH ISPE, Funchal. International Conference on Concurrent Engineering: Research and Applications. pp. 771-778.

McElroy, M. W. (1999), ''The second generation of knowledge management'', Knowledge Management, pp. 86-8.

Malhotra, Y. (1999), ''Beyond 'hi-tech hidebound' knowledge management: strategic information systems for the new world of business'', working paper, BRINT Research Institute.

Nonaka, I. and Takeuchi, H. (1995), The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, *Oxford University Press, New York*, NY.

O'Dell, C. and Grayson, C.J. (1998), ''If only we knew what we know: identification and transfer of internal best practices'', *California Management Review, Vol. 40 No*. 3, pp. 154-74.

OCCS Accessed 2010-05-20 at www.omniclass.org

Olson, G. M. and Olson, J. S. 2000. Distance Matters. Human Computer Interaction. Vol. 15. pp. 139-179.

OWL, Accessed 2010-05-20 at www.w3.org/TR/owl-features/

Protége, Accessed 2010-05-20 at protege.stanford.edu/