Developing an Ontology-Based Team Recommender System using EDON Method: An experience Report

Ma. Celeste Ayub, Ayelen N. Cian, Ma. Laura Caliusco, Emiliano Reynares

CIDISI Research Center – UTN – FRSF – Lavaise 610 – Santa Fe – Argentina
{celeayub, cianayelen}@gmail.com
{mcaliusc, ereynares}@frsf.utn.edu.ar

Abstract. Recently, Team Recommender Systems (TRS) have become extremely common because they are software tools and techniques that helps to organizations to composite team needed to carry out a task requiring multiple skills. TRS have two important problems: (1) managing semantic heterogeneity that occurs when the data describing the same entities related to the real world is represented in different ways, and (2) specialization excess leading to display the objects of highest similarity with the user specified instead of a wide range of options leaving out of consideration the highest possible user interest information. In recent years, the ontology-based information systems have gained the attention of the researchers and practitioners since they handle the semantic heterogeneity problem. Despite of the advance done, building methodologies for developing ontology-based systems is still a research area. In this paper, we report our experience in developing an ontology-based TRS by using the EDON method. The developed TRS analyses human resource information to recommend a work team for a software development project.

Keywords: team recommender systems, ontology-based information system.

1 Introduction

Nowadays, new paradigms for distributed, cross-organizational collaborations are emerging, which e. g. enable open source projects or open innovation. When initiating such projects it can be challenging to choose an appropriate cross-organizational team [1]. The base for finding adequate team members is competence management, which requires a permanently updated knowledge about the employee's capabilities. A team recruiter must know which employee takes care of what task at a certain time, what basic knowledge the employee has and what special knowledge has been acquired. Filling vacant project jobs is rarely based on a structured approach but more or less on personal networking and preferences. That increases the risk that professional criteria are elided and appropriate candidates are overseen. Even when a company makes use of a competence management process it stays a difficult task for the team recruiter to identify the essential necessary skills in many different working domains and to recognize the employees that best fulfill these conditions in a team. So teamplanning requires flexible ways to compare skills in a discrete manner [2].

In order to solve this problem, an ontology-based Recommender Systems (RS) can be built [3]. A RS used to search for persons is called Team Recommender System (TRS). Expert recommendation is an important but also extensively researched problem. In contrast, the generalized problem of team recommendation has not been studied a lot [4]. Different ontology-based approaches have been defined on the algorithmic composition of teams [5-7]. However, each approach bases on its own team composition strategy, making it very difficult to be applicable to other domains.

The objective of this paper is to report an experience on building a TRS by using the EDON Method [8] in the context of an agile software development process. EDON is a method for building from scratch an ontology intended to be used as a structural conceptual model of an information system, encoding business rules in a declarative way.

The remaining of the paper is structured as follows: Section 2 defines the main concepts necessary to understand the content of the paper. Section 3 describes the development of the TRS. Finally, Section 4 is devoted to discussion and lessons learned.

2 Conceptual Foundations

2.1 Team Recommender Systems (TRS)

In the literature, some algorithmic solutions for the problem of team composition based on the use of semantic technologies are already known. In the following, a brief overview on representative work of this field of research is given.

In [5] a semantic based portal is presented for the composition of organizational teams. The user request is formalized as a query, searching the competences required for the task in an ontology used as skills repository. The system returns a set of one or more workers able to cover all the competences required for the task. All the available sets are ranked on the basis of the ontological closeness of query concepts to concepts formalizing skills hold by proposed individuals. In [6] a system integrating the accuracy of concept search with the flexibility of keyword search is proposed to match expertise within academia. The system is based on the use of semantic web technologies and in particular on RDF and XML in order to extract expertise integrated profiles from heterogeneous information sources. In [7] a Description Logics (DL) framework for to the semantic-based composition of teams based on individual skill profiles and on tasks description is presented. In this framework a novel Concept Covering algorithm exploiting the Concept Abduction inference service in DLs was devised. This framework is currently embedded as part of a complete logic-based skill management system.

The above described works show interesting approaches on the algorithmic composition of teams. It can be said that each approach bases on its own team composition strategy, making it very difficult to be applicable to other domains [9].

For this reason in this paper we present our experiences in developing a TRS using a method that allows building a extensible system that can be easily adapted to different contexts.

2.2 Evolutionary Development of ONtologies (EDON)

EDON [8] is an approach for building from scratch an ontology intended to be used as a structural conceptual model of an information system, encoding business rules in a declarative way. EDON adopts a requirement driven, iterative, and incremental approach and it is composed by the processes described next.

Requirements Selection Process. This process is composed by three activities: (1) identification of the functional requirements that involves business rules in their meeting, (2) identification and prioritization of the domain entities involved in the meeting of the requirements identified before, and (3) requirements grouping and selection according to the importance of the entities involved.

Ontology Development Process. This process involves Development Activities that allows evolving from an abstract model toward an computable ontology, and Support Activities are carried out along the whole development process. The Development Activities are: specification, conceptualization, formalization, refinement, implementation and alignment. The Support Activities are: knowledge elicitation and evaluation. This activity classification is based on the Methontology Framework and the techniques to carry out them are based on the different methodologies and good practices for building ontologies that have been developed since mid-1990 [10]. However, EDON considers the performing of the refinement activity with the aim of extending the ontology by focusing on the declarative formulation of business rules.

Ontology Alignment Process. Each application of EDON produces an ontology that supports a disjoint set of functional requirements, i.e., those selected on the specification activity of the iteration. Therefore, the alignment of current and previous version of the ontology is needed as a way to support both set of requirements. Ontology alignment is the process of determining the different types of (inter-ontology) relationships among their terms [11]. As a result, a new ontology composed by sub-ontologies is created. DEs are responsible for establishing the most adequate relationships, given they have such kind of knowledge.

3 Applying the EDON Method for developing a TRS

EDON has been conceived to be intertwined with an iterative and incremental software development process. First, the requirements elicitation activities were performed.

3.1 The roles involved in the Software Development Process

In the development of the TRS three roles were involved: the Software Engineers (SEs), the Domain Experts (DEs) and the Knowledge Experts (KEs). The SEs carried out all the activities necessaries to develop the information systems discussed in Section 2.2. The DE collaborated with the SEs in characterizing the problem domain and the KEs provided knowledge on modeling and evaluating ontologies.

3.2 Requirements Selection

The functional requirements of the TRS were selected considering two issues: (1) the developed ontology should provide ontology-based reasoning over the business rules of the application domain and (2) human users of the TRS should not interact directly with the underlying ontology but through the software application. After defining and analyzing the TRS requirements, the Proyecto and Persona were considered the core entities of the domain as result of the domain entity identification and prioritization activity performing. Finally, a subset of functional requirements mainly involved with such entities were selected for further development. A storyboard exposing a functional requirement belonging to the selected subset is: "*El sistema debe permitirle al usuario seleccionar el puesto para el cual desea obtener postulantes y mostrarle una lista ordenada de los mismos en base al desempeño de la persona.*"

3.3 Ontology Development

Ontology Specification. Once selected the functional requirements to be supported, EDON proposes to stem competency questions (CQs) from such requirements. The CQs were defined taking into account the main entities identified in the previous activity. In addition, a hierarchy of CQs was built. Table 1 and Table 2 show an excerpt of CQs the ontology should be able to answer around the concepts Proyecto and Persona, respectively.

Table 1. An excerpt of CQs around the concept Proyecto.

¿Qué Puestos hay dentro de un Proyecto?

1.1) ¿Cuál es el Perfil requerido para Desempeñar un Puesto dado?
1.2) ¿Cuántos integrantes se requieren en el Proyecto X para el Puesto Y?
1.3) ¿Qué características tiene un Puesto Y para un Proyecto X?
1.4) ¿Qué Director / Sub-Director tuvo el Proyecto X?

Table 2. An excerpt of CQs around the concept Persona.

¿Quién es la Persona?

2.1) ¿Qué información se tiene sobre un candidato seleccionado?
2.2) ¿Qué datos personales se poseen de una persona?
2.3) ¿Cómo es como persona?
2.3.1) ¿Tiene carácter de líder?
2.3.2) ¿Cómo fue evaluado su desempeño para un puesto determinado?
2.4) ¿Qué Proyectos le fueron asignados a una Persona?
2.4.1) ¿Qué Puestos fueron desempeñados por una Persona?
2.4.2) ¿Qué calificación posee una determinada persona por el Puesto desempeñado?

From the CQs a list of the domain entities needed for answering them was identified. An excerpt of this list is shown in Table 3.

Table 3. An excerpt of CQs the ontology should be able to answer

Puesto	Perfil	Director
Sub-Director	Equipo Mantenimiento	Equipo de Desarrollo

Ontology Conceptualization. In this activity, the knowledge about the domain entities is collected from the information sources and its representation is done independently of the modelling paradigm and the implementation language of the target ontology, by using the Lexicon Extended Language (LEL) [12]. During his activity a set of LEL were be built, validated and committed by both DEs and KEs. LEL is a representation of the terminology in the application language, which is classified in four categories: object, subject, verb, and state. Examples of these LELs are shown in Table 4, Table 5 and Table 6.

Table 4. An example of Subject.

DIRECTOR
Noción: -Persona de sexo femenino para Proyectos de tipo A y masculino para los de tipo B, mayor de 18 años a cargo de todo el Proyecto. Requiere haber ocupado el Puesto de Subdirector al menos 1 vez, debe haber tenido un muy buen Desempeño en su Puesto anterior y debe estar habilitado para el Desempeño del Puesto.
Impactos: -Coordina el Proyecto. -Encargado de Convocar a los integrantes de un Proyecto. -Evalúa el Desempeño de los integrantes de un Proyecto. -Organiza reuniones entre los integrantes de un Proyecto. -Tiene a cargo el Equipo de Desarrollo. .Tiene el criterio subjetivo para elegir personas para ocupar futuros Puestos de Trabajo.

Table 5. An example of Object.

PROYECTO
Noción: Conjunto de actividades que se encuentran interrelacionadas y coordinadas por Personas que cumplen un Puesto de Trabajo dentro del mismo o Participan en Proyecto.
Impactos: -Organizado por un grupo de Personas. -La Persona Desempeña un Puesto dentro del Proyecto. -La persona Dicta Curso de Capacitación dentro del Proyecto. -Un Proyecto posee muchos Cursos de Capacitación.

Table 6. An example of Verb.

PARTICIPA EN PROYECTO
Noción: Presencia de una Persona a un Proyecto, Desempeñando o no un Puesto.
Impactos: -Una Persona Participa en Proyecto y Desempeño sobre un Puesto. -Una Persona Participa en Proyecto y no tiene Desempeño sobre un Puesto. -Un Proyecto es Asistido por muchas Personas.

Ontology Formalization. With the aim of formalizing the ontology, the methodology proposed by Breitman and Leite [13] was followed. The main advantages of this methodology over other defined is that it proposes a set of heuristics that can be automatically implemented to build an ontology from a set of LELs. In order to execute this activity, a table that describes each concept was defined. An excerpt of this table that describes the concept Director is shown in Table 7.

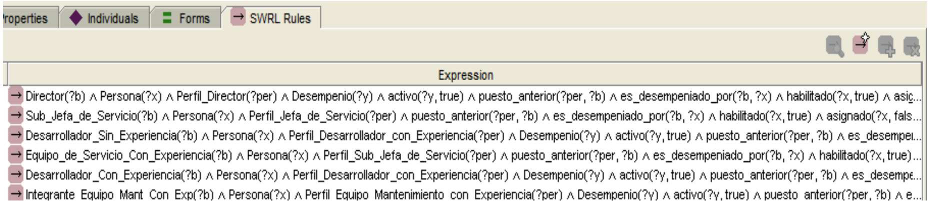
Table 7. The desription of the concept Director.

Concepto	Relaciones	Concepto	Reglas
Director	Director_coordina	Proyecto	-Sexo femenino en Proyectos de tipo A. -Sexo femenino en Proyectos de tipo B. -Edad mayor de 18 años. -Haber sido Subdirector. -Haber tenido muy buen desempeño en su puesto anterior. -Estar habilitado para el desempeño del puesto.
	Director_convoca	Personas	
	Director_evalúa	Desempeño	
	Director_Propone_en_promoción	Personas	
	Director_Tiene_a_cargo	Equipo de Desarrollo	

Ontology Implementation. After the formalization, the ontology was implemented using the free ontology editor called Protégé and the Pellet inference engine that provides sound-and-complete OWL-DL reasoning services[1]. The ontology was written in the OWL-DL 1.0 ontology language and serialized in OWL/RDF format.

Ontology Refinement. The resulting ontology represents the main concepts of the problem domain. The refinement activity consists in further extending the ontology by focusing on the formulation of rules, which are obtained from the knowledge and information sources identified in the specification activity. The rules allow implementing the algorithm for selecting the required team. In literature, different algorithms that consider different aspects for selecting people were defined. The main challenge is how to implement these algorithms in a rule implementation language.

In the case study, the assignment algorithm was implemented according the DE knowledge. The rules were implemented in the Semantic Web Rule Language (SWRL), which provides the ability to express Horn-like rules in terms of OWL concepts [14]. Figure 1 shows the rules implemented in the study case.



Expression

→ Director(?b) ∧ Persona(?x) ∧ Perfil_Director(?per) ∧ Desempenio(?y) ∧ activo(?y, true) ∧ puesto_anterior(?per, ?b) ∧ es_desempeniado_por(?b, ?x) ∧ habilitado(?x, true) ∧ asig...
→ Sub_Jefa_de_Servicio(?b) ∧ Persona(?x) ∧ Perfil_Jefa_de_Servicio(?per) ∧ puesto_anterior(?per, ?b) ∧ es_desempeniado_por(?b, ?x) ∧ habilitado(?x, true) ∧ asignado(?x, fals...
→ Desarrollador_Sin_Experiencia(?b) ∧ Persona(?x) ∧ Perfil_Desarrollador_con_Experiencia(?per) ∧ Desempenio(?y) ∧ activo(?y, true) ∧ puesto_anterior(?per, ?b) ∧ es_desempe...
→ Equipo_de_Servicio_Con_Experiencia(?b) ∧ Persona(?x) ∧ Perfil_Sub_Jefa_de_Servicio(?per) ∧ puesto_anterior(?per, ?b) ∧ es_desempeniado_por(?b, ?x) ∧ habilitado(?x, true)...
→ Desarrollador_Con_Experiencia(?b) ∧ Persona(?x) ∧ Perfil_Desarrollador_con_Experiencia(?per) ∧ Desempenio(?y) ∧ activo(?y, true) ∧ puesto_anterior(?per, ?b) ∧ es_desempe...
→ Integrante_Equipo_Mant_Con_Exp(?b) ∧ Persona(?x) ∧ Perfil_Equipo_Mantenimiento_con_Experiencia(?per) ∧ Desempenio(?y) ∧ activo(?y, true) ∧ puesto_anterior(?per, ?b) ∧ e...

Fig. 1. Rules implemented in SWRL.

At the moment of building the rules, some relations and properties are not used. Following the conciseness design principle, the ontology can be improved by deleting

[1] Support, downloads and documentation about the integration of Protégé editor and Pellet inference engine can be found in http://protege.stanford.edu/

these elements. Conciseness refers to if all the information gathered in the ontology is useful and precise [15].

In addition, when defining rules some properties of the classes and additional classes are added. The classes that are added to the ontology are *deduced classes* that mean they are classes whose instances are deduced as result of the rule execution.

3.4 Ontology Alignment

The alignment activity consists in establishing a set of correspondences between entities belonging to two different ontologies. As a result, a new ontology composed by sub-ontologies is created. Subsections above have depicted the development of the first version of the ontology of the study case, which does not involve the performing of alignment activities. During the development of the TRS a second iteration was performed and an alignment process was executed the Protégé-based PROMPT tool. The resulting ontology can be found in http://code.google.com/p/proyecto-ayub-cian/.

3.5 Ontology Evaluation

Ontology quality was assessed by means of OQuaRE [16], which is a framework for ontology quality evaluation based on the SQuaRE standard for software quality evaluation[2]. OQuaRE considers ontologies as artefacts obtained by means of a building process and evaluate them independently of any particular development process.

OQuaRe defines a quality model and quality metrics for ontology evaluation. Quality model is divided into a series of characteristics organized into subcharacteristics which are evaluated by applying a set of metrics. OQuaRE defines the criteria to transform the quantitative scores of each metric into the range 1-5 and establishing that 1 means not acceptable, 3 is minimally acceptable and 5 exceeds the requirements. Score for each subcharacteristic is the mean of its associated metrics and the score of each characteristic is the mean of its subcharacteristics. The set of characteristics scores enables the identification of strengths and flaws of an ontology. Characteristics evaluated for the ontology developed using EDON were:

- *Structural* characteristic involves formal and semantic properties that are important when evaluating ontologies since it accounts for quality factors such as consistency, formalization, redundancy or tangledness.
- *Functional adequacy* characteristic refers to the appropriateness of the ontology for its intended purpose, according to the categories identified by [17].
- *Maintainability* feature is related to the capability of the ontologies to be modified for changes in the environment, in requirements or in functional specifications.
- *Compatibility* characteristic refers to the ability of two or more ontologies to exchange information and/or to perform their required functions while sharing the same hardware or software environments.

[2] International Organization for Standardization (ISO) ISO/IEC 25000 2500 Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE)

- *Transferability* characteristic is the degree to which the ontology can be transferred from one environment to another.
- *Operability* characteristic refers to the effort needed for use the ontology and, in the individual assessment of such use, by a stated or implied set of users.
- *Reliability* dimension is the capability of the ontology to maintain its level of performance under stated conditions for a given period of time.

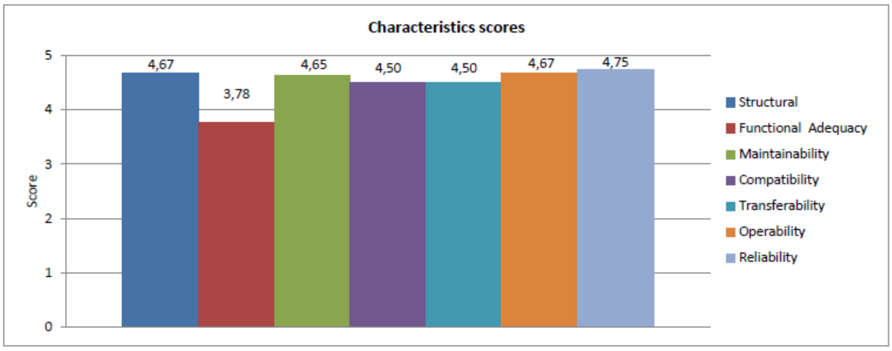


Fig. 2. Characteristics scores of the ontology developed.

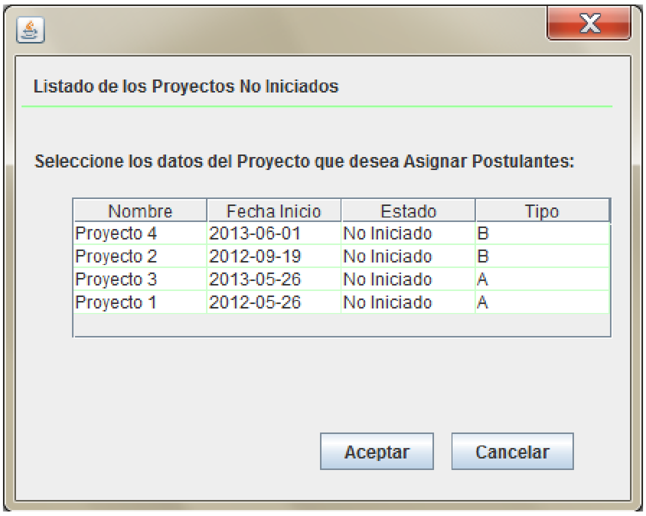
Figure 2 highlights the level of quality of the ontology. According to OQuaRE, the ontology outperform the minimally acceptable quality in all dimensions. Mean global score for ontology quality is 4.50, very close to the optimal quality.

3.6 TRS Implementation

After the ontology evaluation, the TRS was implemented in Java by using the JENA framework. Business logic is implemented by means of the rules of the ontology, improving the maintainability quality criterion of the system. The resulting system was evaluated in conjunction with the DE. The system code can be found in http://code.google.com/p/proyecto-ayub-cian/. The system quality from the user requirement perspective was evaluated by the expert of the domain.

The TRS implemented provides a couple steps for complete a work team by selecting persons who will be develop the work necessary to be done for a particular Project (Proyecto) according to the criteria of this organization.

The first step is selecting the Project (Figure 3). For that goal, the system provides you an interface, this interface expose the Projects in state ‘No Iniciado’ (this means, the system cannot assign people in a Project already in progress). This is a requirement from the company, it cannot start a Project without all people available to start.

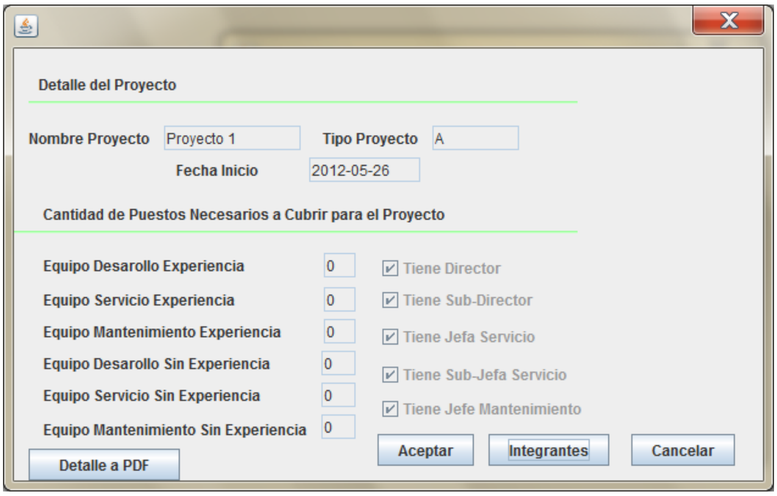


Listado de los Proyectos No Iniciados

Seleccione los datos del Proyecto que desea Asignar Postulantes:

Nombre	Fecha Inicio	Estado	Tipo
Proyecto 4	2013-06-01	No Iniciado	B
Proyecto 2	2012-09-19	No Iniciado	B
Proyecto 3	2013-05-26	No Iniciado	A
Proyecto 1	2012-05-26	No Iniciado	A

Fig.3. The TRS implementation for Select a Project.



Detalle del Proyecto

Nombre Proyecto: Proyecto 1
Tipo Proyecto: A
Fecha Inicio: 2012-05-26

Cantidad de Puestos Necesarios a Cubrir para el Proyecto

Equipo Desarollo Experiencia	0	☑ Tiene Director
Equipo Servicio Experiencia	0	☑ Tiene Sub-Director
Equipo Mantenimiento Experiencia	0	☑ Tiene Jefa Servicio
Equipo Desarollo Sin Experiencia	0	☑ Tiene Sub-Jefa Servicio
Equipo Servicio Sin Experiencia	0	☑ Tiene Jefe Mantenimiento
Equipo Mantenimiento Sin Experiencia	0	

Fig.4. The TRS implementation Selecting a Project for Fill all the jobs.

In the interface shown in Figure 4, when click in ‘Aceptar’ (Accept), the Project is selected for finding the people needed to start this Project.

The second step is selecting the job who is not yet occupied by another one (Figure 5). The system has two types of works, works who requires only one person like 'Director' (Principal), 'Sub-Director'(Co-Principal), 'Jefa de Servicio' (Service Chief), 'Sub-Jefa de Servicio' (Service Assistant of Chief) and 'Jefe de Mantenimiento' (Maintenance Chief). On the other hand, there exists some works that has to be performed by more than one person, e.g. 'Equipo de Desarrollo' (Team Development), 'Equipo de Servicio' (Team Service), 'Equipo de Mantenimiento' (Maintenance Team).

The system allows selecting people according to de job bases. The postulant are selecting by the OWL Rules in the ontology (Figure 1). That rules are checking with the expert of domain.

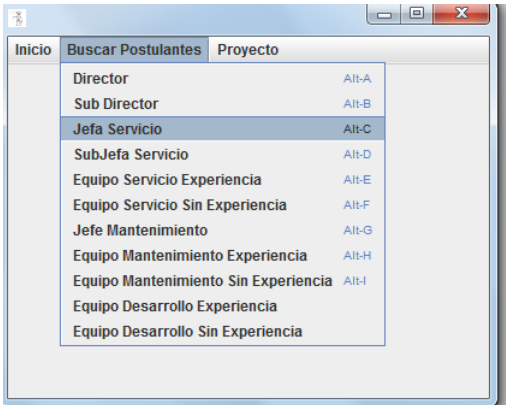
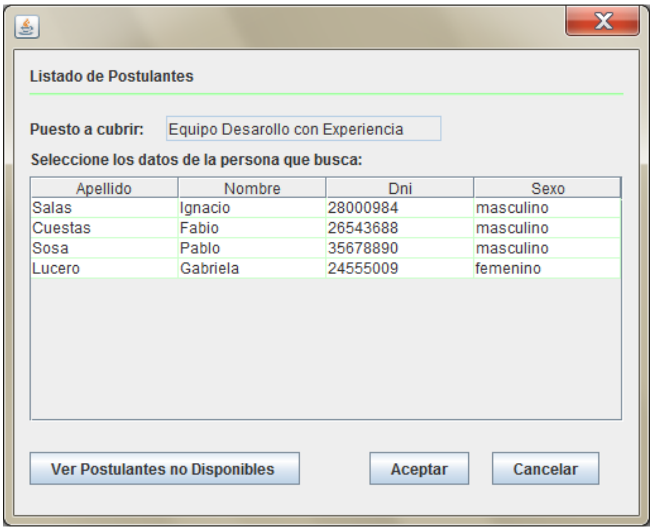

Inicio
Buscar Postulantes
Proyecto
Director Alt-A
Sub Director Alt-B
Jefa Servicio Alt-C
SubJefa Servicio Alt-D
Equipo Servicio Experiencia Alt-E
Equipo Servicio Sin Experiencia Alt-F
Jefe Mantenimiento Alt-G
Equipo Mantenimiento Experiencia Alt-H
Equipo Mantenimiento Sin Experiencia Alt-I
Equipo Desarrollo Experiencia
Equipo Desarrollo Sin Experiencia

Fig.5. Selecting a Job.

Then the system shows a list from the all postulant, let the user find who will be selected for this job. This way simplifies the really hard work who represents selecting people inside an organization. If a requirement of a job change, the user can modify the OWL Rule, changing the decision of the system, without review all the history of work for everyone who works in the organization.



Listado de Postulantes

Puesto a cubrir: Equipo Desarollo con Experiencia

Seleccione los datos de la persona que busca:

Apellido	Nombre	Dni	Sexo
Salas	Ignacio	28000984	masculino
Cuestas	Fabio	26543688	masculino
Sosa	Pablo	35678890	masculino
Lucero	Gabriela	24555009	femenino

Fig.7. The TRS implements Select a Postulant for the job Development Team with Experience.

In order to select a person, click on ‘Aceptar’ (Accept). The system lets the user ask the postulant if he wants to do this job. If the postulant accepts, the system updates the ontology with the new state and the user can ask to another person for complete the Team. If the postulant does not accept, the system does not update the ontology. This cycle is repeated until the team is formed. This steps is the same for all the jobs in the Project.

3.7 TRS Testing

The goal of the TRS testing activity was to verify its functionality by means of their external interfaces, with the aim to assess the system quality by performing a verification process of the requirements established on early stages of the system development. This approach facilitates the execution of testing procedures allowing to obtain all the possible combinations of testing cases for an specific requirement. As a result, a testing script is generated which can be executed at the end of each iteration of the development process.

4 Discussion and Lessons Learned

In this paper we have reported our experience and showed the satisfactory results in developing an ontology-based Team Recommender Systems using the EDON Method in the context of an agile software development process.

The EDON Method defines activities for building from scratch an ontology considering the functional requirements. When you begin to design the system and its components is essential to prioritize the requirements and know what stage will incorporate every improvement. The requirement prioritization in our system helped us to define the boundaries of each iteration by prioritizing the requirements oriented to the ontology.

Then, EDON proposes to develop an ontology that fulfills the requirements of the development cycle to which it belongs. From requirements, through CQs and LELs, you get the necessary information about the domain which is then captured as objects, relationships and properties in the implemented ontology. With regard to CQs, they can lead to create objects, relations or properties that are not relevant to the system, but they are for the environment in which the system is embedded. This happened to us in our development and is mainly due to those who are not familiar with the development of ontologies think in terms of the system. With regards to LELs, the heuristics used to built the ontology from them is not complete enough. Then, although the ontology conceptualization by using CQ and LELs has proven to be useful to facilitate the communication among the DEs, SEs and KEs, we consider that a more powerful formalism will improve the way complex business rules are expressed. With regards to LELs and the heuristics used to formalize the ontology, it is important to highlight that such proposal define a systematic process that results easy to understand and apply. According to our knowledge at the time of developing the system, there was not another proposal defining a process with such features. However, although the process has been useful to facilitate the communication among the DEs, SEs and KEs, we consider that a more powerful formalism will improve the way complex business rules are expressed. As a consequence, it will be necessary to formulate a new process to generate the ontology structure from such high level expressions.

Modeling errors committed were offset during the ontology refinement stage. This step is extremely important to develop a quality ontology, which has the necessary data to represent the knowledge of the system and meets the ontology quality criteria.

While the method proposed perform EDON first stage of refinement, then the implementation phase, in our case study we implemented a first version of the ontology and then did the refinement of ontology implemented. Thus, the refining step was made with support from the evaluation activity.

As a final step EDON proposed to align the ontologies that are developed throughout the history of the system. Therefore this methodology allows and provides the system to grow. This feature is a very important point for the current systems that must constantly adapt not only to changes in technology, but also to the new requirements imposed by the users because their needs are becoming increasingly specific.

As regards TRS, to build a specialized filter, based on a user profile or the data it needs, you can perform a search based on this profile. One of the advantages to be gained by using this type of system is defined Project Requirements Profile allowing it to find people based on compliance with the specified considering the Person Competence Profile. Using SWRL rules could filter out individuals who meet these requirements. SWRL rules are powerful tools for implementing teamplanning algorithms.

Future work will be devoted on improving the EDON method taking into account the lesson learned, and studying how to implement teamplanning algorithms in SWRL.

Acknowledgments

The authors are grateful to Mariano Rubiolo for insight and feedback about domain approached by the developed Team Recommender System.

5 References

1. Brocco, M.; Hauptmann, C.; Andergassen-Soelva, E. Recommender System Augmentation of HR Databases for Team Recommendation. In Proc.: 22nd International Workshop on Database and Expert Systems Applications, pp 554-558, (2011).
2. Gottwald, S.: Planning Teams with Semantic Web Technologies. GI-Edition Lecture Notes in Informatics “Informatik 2009 - Im Focus das Leben", Lübeck. (2009).
3. Keim, T.: Extending the Applicability of Recommender Systems: A Multilayer Framework for Matching Human Resources. In Proc.: 40th Annual Hawaii International Conference on System Sciences, (2007).
4. Datta, A.; Yong, J.; Ventresque, A.: T-RecS: Team Recommendation System through Expertise and Cohesiveness. In Proc: WWW Conference Committee, pp. 201-204, (2011).
5. Hefke, M. and Stojanovic, L.: An ontology-based approach for competence bundling and composition of ad-hoc teams in an organization. In Proc.: I-KNOW’04, pp 126–134, 2004.
6. Liu, P. and Dew, P.: Using semantic web technologies to improve expertise matching within academia. In Proc.: I-KNOW ’04, pages 370–378, (2004).

7. Colucci, S.; Di Noia, T.; Di Sciascio, E.; Donini, F.M.; Piscitelli, G. and Coppi, S.: A formal approach to ontology-based semantic match of skills descriptions. In Proc.: 2005 ACM symposium on Applied computing, New York, NY, USA: ACM, pages 1314-1319, (2005)
8. Reynares, E.; Caliusco, Ma. L.; Galli, Ma. R.: "EDON: A Method for Building an Ontology as Software Artefact". In Proc.: 41° Jornadas Argentinas de Informática en el Simposio de Ingeniería de Software, (2012).
9. Brocco, M., Hauptmann, C., Andergassen-Soelva, E.: Recommender System Augmentation of HR Databases for Team Recommendation. In: DEXA Workshops(2011)554-558.
10. Gómez-Pérez, A. and Fernández-López, M. and Corcho, O.: Ontological Engineering. Springer/Heidelberg. (2004).
11. Pavel, S and Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. In IEEE Transactions on Knowledge and Data Engineering, PP-99 (2011).
12. Breitman, K. and Leite, J.C.S.P.: Lexicon based ontology construction. In Software Engineering for Multi-Agent Systems II.. Springer/Berlin/Heidelberg LNCS 2940(2004) 41-45.
13. Leite, J.C.S.P. and Franco, A.P.M.: A Strategy for Conceptual Model Acquisition. In Proc. IEEE International Symposium on Requirements Engineering. IEEE Computer Society Press. 243-246. (1993)
14. O'Connor, M. Knublauch, H., Tu, S. & Musen M.: Writing rules for the semantic web using SWRL and Jess. In: Proc. 8th Int. Protégé Conference, Protégé with rules, (2005).
15. Gómez-Pérez, A. Ontology evaluation. In Steffen Staab and Rudi Studer, editors, Handbook on Ontologies, First Edition, chapter 13, pages 251-274. Springer, (2004).
16. Duque-Ramos, A. and Lopez, U. and Fernandez-Breis, J. T. and Stevens, R. and Aussenac-Gilles, N.: OQuaRE: a SQuaRE-based approach for evaluating the quality of ontologies. Journal of Research and Practice in Information Technology, 43, 159-173 (2012).
17. Stevens, R. and Wroe, C. and Gobel, C. and Lord, P.: Application of ontologies in bioinformatics. In Handbook of Ontologies in Informations Systems. Staab, S. and Studer, R. (eds). pp 635-658. Springer. (2008).