Expert Systems with Applications 38 (2011) 1160–1170

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Supporting small teams in cooperatively building application domain models

Cesar Augusto Tacla [a,*], Ademir Roberto Freddo [a], Emerson Cabrera Paraiso [b], Milton Pires Ramos [c], Gilson Yukio Sato [a]

[a] Federal Technological University of Paraná (UTFPR), Curitiba, Brazil
[b] Pontifical Catholic University of Paraná (PUCPR), Curitiba, Brazil
[c] Paraná Institute of Technology (TECPAR), Curitiba, Brazil

ARTICLE INFO

Keywords:
Knowledge engineering
Collaborative systems
Multiagent system
Folksonomy
Ontology

ABSTRACT

Building application domain models is a time-consuming activity in software engineering. In small teams, it is an activity that involves almost all participants, including developers and domain experts. In our approach, we support the knowledge engineering activity by reusing tagging done by team participants when they search information on the Web about the application's domain. Team participants collaborate implicitly when they do tagging because their individually created tags are collected and form a folksonomy. This folksonomy reflects their knowledge about the domain and it is the base for eliciting domain model elements in the knowledge acquisition and conceptualization tasks in a consensual way. Experiments provide evidence that our approach helps team participants to build richer domain models than if they do not use our software tool. The tool allows the reuse of simple annotations as long as users learn about the application's domain.

1. Introduction

Domain modeling is an activity present in a great number of software development (SD) projects. It is essentially a cooperative activity and hence an interesting field of experimentation for computer supported cooperative work (CSCW). It is often the case that developers do not know deeply the domain of the application. In order to learn, elicit knowledge to be represented, build a conceptual vision of such knowledge and, finally, formalize and represent it in a machine-readable format, developers use several sources of information, including the experts of the domain. For instance, a SD team developing a domain model for a tourism website has to interview people working in this business and its potential users, search for information in tourist guides, travel agencies and tourist offices learning which elements are relevant for designing the system's use cases and how such elements are related to each other.

This is a long process since knowledge is learned in an incremental way and people should commit to a certain view of the domain. Different people have different views of the same piece of the domain being modeled and resulting conflicts must be managed. Consequently, there is a great difference from versions of the domain model built in the initial phases to the ones at the end of the project.

In a SD process, application domain models may take different forms such as entity-relationship models, conceptual-level schemas, object-oriented data models (OODM) and ontologies, the latter being adopted as reference in this work. We adopted ontologies because they are built consensually and reaching such a consensus demands cooperation as stated by Gruber (1993): "an ontology is a formal, explicit specification of a shared conceptualization". Guarino and Giaretta (1995) consider an ontology in the artificial intelligence (AI) sense as "a logical theory which gives an explicit, partial account of a conceptualization". Ontologies can be represented taking AI-based approaches (e.g. frames), logics, among others. In this work, we assume ontologies are represented with Description Logics (Baader, McGuinness, Nardi, & Patel-Schneider, 2003). In this logic, the representational primitives are concepts (classes of objects), roles (binaries relations between concepts), and individuals (instances of classes).

According to (Borgida & Brachman, 2003), all of the above mentioned kinds of application domain models are comparable since they rely on an object-centered view of the world including notions like individual objects, which have relationships to each other, and which are grouped into classes. Even notational languages used to represent OODM and ontologies can be the same, such as the UML (*Unified Modeling Language*), the similarity among different kinds of domain models was also noted by Studer, Benjamins, and Fensel (1998) when they present the communalities of Software Engineering and Knowledge Engineering.

* Corresponding author. Address: Avenida Sete de Setembro, 3165, 80230-901 Curitiba, Brazil. Tel.: +55 41 3310 4685.
E-mail address: tacla@utfpr.edu.br (C.A. Tacla).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.05.009

Nowadays, there is an intense research in emergence of knowledge from social applications like BibSonomy,[1] Semantic-MediaWiki,[2] and Richards (2009) proposing new methods and techniques to acquire knowledge. In this work, we apply one of these techniques in small SD teams to support knowledge engineering activity by acquiring knowledge from folksonomies.

In social applications, a personomy is produced by user tagging activity in any social tagging system and it is formed by all tags of the user represented in a plain structure. The collection of personomies constitutes a folksonomy (Jaschke, Hotho, Schmitz, Ganter, & Stumme, 2008). The folksonomy is dynamic as long as users learn new things and review their personomies, including and excluding their tags. Users can annotate resources with different tags depending on their social or cultural backgrounds, expertise and perception of the world (Begelman, Keller, & Smadja, 2006; Golder, 2005; Peterson, 2006; Wu, Zubair, & Maly, 2006) and thus it is an important source of information.

Typically, small teams in SD have fewer than 10 members, and wasteful activities can be detrimental or even lead the development effort to failure. Pollice, Augustine, Lowe, and Madhur (2004) describe a SD project in which members of the development team began by identifying artifacts they felt were absolutely required, including the vision statement, risk list, development case, use cases, test cases, project plan, glossary, and architecture. As the development effort progressed, the team created many of these essential artifacts informally. For example, a test plan might be simply a sticky note attached to the wall. As the team members stated, it was "good enough" for the team's needs. By creating many of their artifacts according to this principle, they avoided waste time and attention on documentation that was not necessary for delivering a stable and well-tested product.

Application domain modeling in small teams of SD is an activity that involves practically all participants. Such activity is one among several they develop day-to-day and it could be just partially supported by computers because there are a considerable amount of face-to-face discussions in order to build a shared conceptual view. Thus, the problem is how to help developers to build such models in a cooperative way during the project. The cooperative modelling includes (i) coping with frequent modifications on the information sources and their influence in the current version of the domain model, (ii) reaching consensus on the knowledge representation structure, and (iii) tracing modeling decisions.

The general goal of our work is to support different types of activities in SD for small teams using a multiagent system (MAS). The focus of this article is specifically on the domain modeling activity. We show evidences that a folksonomy built from personomies can help a small team to build better ontologies than if they work without remembering simple annotations made when they visit web pages about an application domain. The folksonomy results from implicit collaboration given that users tag resources individually.

This paper presents a review of the literature on CSCW for domain modeling, the fundamentals on ontology development, ontology learning, ontology evolution and multiagent systems, the proposed multiagent architecture highlighting the domain modeling activity, the experiments in order to evaluate the level of support offered to the users, analysis of the results, and finally, a conclusion.

2. Related work on CSCW for domain modeling

Research on collaborative construction of knowledge has been motivated by the difficulties related to knowledge engineering activity, mainly that concerning knowledge acquisition and formalization. There are a number of applications that provide functionalities allowing users to build taxonomies, ontologies and to populate them such as Collaborative Protégé (Tudorache et al., 2008) and SOBOLEO.[3] These applications provide functionalities related to versioning, collaborative editing, and asynchronous communication for annotating changes on the elements of the ontology being collaboratively edited.

Sure et al. (2002) created an environment named OntoEdit to build ontologies collaboratively. The environment embodies a methodology to develop ontologies and provides inference and collaboration functionalities. OntoEdit supports the three phases of such ontology development methodology: requirements specification, refinement and evaluation. In all the phases, ontology engineers, domain experts and users can collaborate. All members of the ontology development team can modify and extend the ontology. They also are informed of such modifications, allowing them to monitor the evolution of the development. OntoEdit also provides mechanisms to guarantee some degree of consistency and concurrency. To ensure consistency, a mechanism to lock concepts, instances and relations was implemented. Using such a mechanism, team members can also lock parts of the ontology and assign different members to work in each part.

Schaffert, Gruber, and Westenthaler (2005) present the idea of "semantic wikis" as a tool to allow collaboration among domain experts and developers working on ontology development. The system also aims at improving searching and navigation by using semantic annotation. They consider that "semantic wikis" could facilitate the participation of non-technical users, the evolution of the knowledge in the wiki, and the collaboration among users. Domain experts could easily make explicit knowledge relevant to the ontology using a wiki. Developers should help them if a more formal representation is required.

Zhdanova (2008) presents a bottom-up approach to build ontologies based in communities using portals. The idea is to improve the services provided by the portal by using semantic web applications. In this approach, community members build an ontology that is potentially more relevant for the community, less expensive and easier to update. The ontology is organized as three different levels: individual level, community level and portal level. Data from the individual level is used to build the community level and data from the community level is used to build the portal level. In such a process, a mechanism to achieve consensus is employed. The system also counts on a mechanism to represent a social network in which links among users are established semantically. Zhdanova (2008) uses a bottom-up approach (folksonomies) that integrates individual profiles and domain representation in order to provide better retrieval service to her users through a portal.

Dotsika (2009) argues that ontologies and folksonomies perform a similar role in web content classification schemes, although being engineered in different ways. She proposes some requirements for a methodology to conciliate both classification approaches and guarantee the quality of the final information model. Dotsika discusses various existing methodologies to integrate ontologies and folksonomies to conclude that they are still in an experimental phase, they lack automation and they do not address certain quality issues. She identifies as requirements for a methodology to integrate folksonomies and ontologies, the quality issues, semantic enrichment, mapping completeness, trust and ethics.

[1] http://www.bibsonomy.org.

[2] http://semantic-mediawiki.org/.

[3] http://www.soboleo.com.