Advanced Engineering Informatics 30 (2016) 337–353

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

An ontologist feedback driven ontology evolution with an adaptive multi-agent system

Souad Benomrane [a,*], Zied Sellami [b], Mounir Ben Ayed [a]

[a] REGIM-Lab.: REsearch Groups in Intelligent Machines, University of Sfax, ENIS, BP 1173, Sfax 3038, Tunisia
[b] LIP6 – Equipe ACASA, 4 Place Jussieu, Boite Courrier 169, 75252 Paris Cedex 05, France

ARTICLE INFO

Article history:
Received 27 October 2015
Received in revised form 8 May 2016
Accepted 10 May 2016

Keywords:
Adaptive multi-agent system
Ontology evolution
Ontologist feedback

ABSTRACT

In a dynamic environment, it is necessary to make changes to an ontology according to new knowledge and user needs. However, ontology evolution is still a complex and time-consuming task. In this paper, we presented OntoAMAS, an ontologist feedback tool based on an adaptive multi-agent system (AMAS) for ontology evolution. It consists of two components: (i) an AMAS (concept and term agents) that represents the current state of an ontology and (ii) a graphical interface which allows to manage the different interactions between the ontologist and AMAS proposals. First, we defined an adaptive behavior that enables agents to react to the ontologist's feedback. The ontologist gives his/her feedback (elementary and composite changes). He/She can also add new terms and concepts. Then, the AMAS self-organizes and produces an updated ontology with new proposals. It works in an interactive and iterative way until a satisfactory state of the ontology is achieved. Second, we proved that OntoAMAS guarantees that the adaptive skills we added to agents allow them to detect the uselessness of some proposals so as to avoid them together with the wrong ones and to propose others. The experimental results show the relevance of OntoAMAS and the effectiveness in time performance of its Protégé (ontology editor) GUI.

1. Introduction

Ontologies developed and used for various applications have been increasing over the years. The major issue faced in ontologies is their change or evolution. Ontology evolution seeks to keep an ontology up to date with the changes of the domain it models, to new knowledge and user needs. These changes should be implemented in the ontology and should be managed in such a way that ensures to safeguard its consistency, structure and continuity across different versions of the ontology. Ontology evolution is a process that involves a number of steps, for which many techniques, tools and approaches might be needed [1]. In the literature, the majority of approaches tried to automatically perform one or two steps of the process to simplify the user's (ontologist's) task. However, throughout the evolution process stages, the ontologist was immediately involved. The ontologist chooses to apply the changes in the ontology, places the new entities (concepts, terms, relations), etc. However, manually evolving an ontology is a costly, complex, and time consuming process.

To the best of the authors knowledge, there are only two systems proposing an automatic ontology evolution namely EVOLVA and DYNAMO. The first [2] is only efficient for the evolution of English ontologies and has difficulties when the concerned domain is very specific. The second system was proposed by Ottens [3] as a first prototype of DYNAMO and then a new tool, called ≪ DYNAMO-MAS ≫ [4], was developed. DYNAMO prototype is based on an adaptive multi-agent system (AMAS) to co-construct and evolve ontology from texts. However, experiments [4] showed that linguistic clues are insufficient and claim that for the work to be more effective the intervention of an ontologist is required. An ontologist is a cognitive engineer using expert interviews and information from texts to construct ontologies.

In conclusion, ontology evolution needs to be more automated without ignoring the role of the ontologist especially during the validation step [5,6]. In this paper we aimed to answer two main questions: (i) How to improve the quality of evolution proposals (changes) by referring to the ontologist's feedback? (ii) How exploiting of the ontologist's feedback may reduce his frequent involvement? Therefore, our objective was to improve the results proposed by the AMAS by exploiting the ontologist's feedback. Consequently, improving the performance of AMAS may enable to reduce the frequent involvement of the ontologist.

* Corresponding author.
E-mail address: souad.benomrane@ieee.org (S. Benomrane).

http://dx.doi.org/10.1016/j.aei.2016.05.002
1474-0346/

By focusing on finding answers to solve the above mentioned challenges in ontology evolution, this paper contributes to the following key issues in two research areas: (a) Ontology Engineering, by proposing an extended approach [7] of an earlier work [4] in ontology evolution. Indeed, some improvements were achieved in the DYNAMO-MAS approach to better evolve the ontology by implementing some existing adaptive rules [4] and extending it with new adaptation behaviors added to the concept and term agents; (b) AMAS, by developing a tool called ≪ OntoAMAS ≫ that can be triggered when a new document is added to the corpus at any time or when the ontologist makes some improvements in the proposed results. OntoAMAS and the ontologist modify the same ontology in a cooperative and adaptive way: this process relies heavily on the strong relation between the action of one of them and the reaction of the other. The adaptation mechanisms added to the agents behaviors enable them to exploit the ontologist's feedback (elementary and composite changes) and self-adapt to personalize the OntoAMAS proposals. That is to say, we aimed to exploit the ontologist's feedback to personalize OntoAMAS proposals.

The remainder of this paper was organized as follows: Firstly, several related works on the existing approaches for evolving ontologies were described in Section 2, justifying the choice of AMAS for the management of ontologies and discussing the limitations of the current version of DYNAMO that led to the creation of OntoAMAS. Section 3, was devoted to the description of the OntoAMAS approach. The implementation of OntoAMAS was presented and the results were discussed in Section 4. Finally in Section 5, the paper major conclusions were drawn and some future works were suggested.

2. Related works

2.1. Ontology evolution

In the literature, ontology evolution appears as a part of a global scope of ontology maintenance. Stojanovic [1] defines ontology evolution as a process to ≪ adapt and change the ontology in a timely and consistent manner ≫. The process of ontology evolution contains 6 steps: (a) change capture, (b) change representation, (c) change semantics, (d) change implementation, (e) change propagation and (f) the validation. In order to manage these different tasks, several approaches propose: (i) tools and guidelines to capture change needs [8] or to identify new knowledge leading to some changes in the ontology [9,10]; (ii) models to represent these changes [1,11]; (iii) rules to identify the change semantics to avoid semantic inconsistencies that could occur as a result of these changes [1,12]; (iv) tools that implement changes [1,13]; (v) other studies ensure the propagation of changes and the update of applications and artefacts connected to the modified ontology [9,14,15,11]. Ultimately, evolution support tools and versioning were subject of discussion of many other approaches [13].

Several approaches and tools have been proposed in the literature [16]. Each tool seeks to automate one or two stages of the process. Stojanovic [1] was interested in a change representation phase where changes are represented following a specific model. This representation is followed by the change semantics phase, during which syntactic and semantic inconsistencies could appear as a result of changes. The **KOAN** *ontology evolution* tool [1] leads the formulation of changes by suggesting ontology improvements. Klein [9] and Rogozan [15] were rather interested in ontology versioning to manage the changes propagation in order to ensure the consistency of the underlying ontology and all dependent artefacts. Rogozan was inspired by the work of Klein and Stojanovic to propose an approach and a tool to manage the ontology evolution and versioning. To implement this approach, Rogozan [15] provides a tool consisting of two modules: **ChangeHistoryBuilder** and **SemanticAnnotationModifier**. Djedidi [12] focused on the steps of semantics of change and change validation. She proposes an approach and a prototype of evolution (**Onto-Evo^{a}l**: Ontology Evolution-Evaluation). It is an automated process driving the change application while maintaining the evolved ontology consistency. In addition, the tool integrates an evaluation activity supported by a defined ontology quality model. This model is used to guide inconsistency resolution by assessing the impact of the resolutions proposed by the evolution process on ontology quality and selecting the resolution that preserves the quality of the evolved ontology. Luong [14] focused on the stage of change propagation to the dependent artefacts namely semantic annotations. Tissaoui [11] focused on change representation step. He proposed an approach and a tool (**EvOnto**) that supports a coherent joint change management of (termino-ontologies or TOR) and semantic annotations by anticipating all the consequences of a change on the TOR and on the annotations. This allowed avoiding missing some of the impacts of a change.

It is worth noting, however, that the identification of change and changes representation are manual. The ontologist is often the one who is in charge of detecting the need to change the ontology and expresses this evolution. In general, throughout the evolution process stages, the ontologist was involved in each step. The ontologist chooses to apply the changes in the ontology, places the new entities (concepts, terms, relations), etc. It is a laborious process that requires a lot of time and effort. To minimize the frequent involvement of the ontologist, two solutions proposing an automatic ontology evolution from texts were addressed by two systems: **EVOLVA** [2] and **DYNAMO** [3].

EVOLVA [2] contributes in a unique way to the ontology evolution by analyzing the existing domain data, and being based on online ontologies as source of background knowledge for the enrichment step. Indeed, EVOLVA relies on these existing ontologies to seek a relation between the entity to be added to the ontology and the current ones already belong to the ontology. EVOLVA is relevant for the English ontologies since the majority of available ontologies on the Web are represented in English. Therefore, EVOLVA is less useful to enrich French ontologies. Experiments have also shown some limitations in EVOLVA when handling a very specific domain [5]. It has difficulties in detecting relations between a new entity and the already existing ones in the ontology.

The second system is the first prototype of DYNAMO [3] (an acronym of DYNAMic Ontologies). It is not able to evolve the ontology but it allows its construction from scratch. DYNAMO is based on an AMAS to construct and maintain an ontology. The agents of the AMAS implement a distributed clustering algorithm to identify clusters of terms from a large corpus of texts and organize these clusters into a set of concepts in a hierarchy. Each agent represents an extracted candidate term. Thanks to the statistical features, similar terms move closer to create and position concepts. The MAS evolves until all agents are hierarchically linked. The final state of the MAS corresponds to the ontology. The ontologist can then validate or reject. Experimentations carried out with this first prototype of DYNAMO confirmed the inefficiency of statistical approaches [5] when dealing with short texts.

To overcome the previous limitations mentioned in EVOLVA and in the earlier prototype of DYNAMO, DYNAMO-MAS [5] was proposed not as an evolution of the first prototype but as a new approach based on AMAS and using linguistic and statistical criteria. As our proposed approach is an evolution of DYNAMO-MAS, the next section is devoted to justify the choice of AMAS to represent the ontology and to present an overview of the second prototype of DYNAMO.