An Adaptive Multi-Agent System for Ontology Co-evolution

Souad Benomrane[1], Zied Sellami[2], Mounir Ben Ayed[1] and Adel M. Alimi[1]
[1]*REGIM-Lab.: REsearch Groups in Intelligent Machines, University of Sfax, ENIS, BP 1173, Sfax, 3038, Tunisia*
[2]*LIP6 - Equipe ACASA, 4 Place Jussieu, Boite Courrier 169, 75252 Paris Cedex 05, France*

Keywords: Dynamic Ontology, Co-Evolution, Adaptive Multi-Agent System.

Abstract: A dynamic ontology evolution reflects the ontology adaptation, to a set of changes and their propagation to the other dependent components, to ensure its consistency. This process needs a frequent involvment of the user (ontologist), which is a complex and time consuming task. As a solution, in this paper we present an extension of an ontology evolution tool called DYNAMO MAS based on an adaptive multi-agent system (AMAS). We improve agents by adding new behaviour to adapt to ontologist actions in order to improve the proposals already made and to propose others.

1 INTRODUCTION

The number of ontologies that are developed and used for various applications are in increase. The major problem faced in ontologies is their change or their evolution. Ontologies can change for various reasons, for example when user needs change or when the knowledge of the domain evolve. Sometimes, the original representation needs some conceptualization adjustment.

Ontologies must be modified regularly while maintaining their coherence, to adapt to the evolution of field to which they relate. Thus, (Stojanovic, 2004) defines ontology evolution as *a timely adaptation to the arisen changes and the consistent propagation of these changes to dependent artefacts*.

In the litterature, several approaches have been proposed to manage the process of ontology evolution, (Stojanovic, 2004) (Klein, 2004) (Luong, 2007) (Djedidi and Aufaure, 2010) (Tissaoui et al., 2011) (Sellami et al., 2012). All these proposals are based on the approach of Stojanovic (Stojanovic, 2004), which is the first that proposed a process of ontology evolution. It contains six steps: (1) change capturing, (2) representation phase, (3) semantics of change, (4) the implementation of change, (5) change propagation and (6) the validation phase. Throughout these steps, the ontologist was involved in every stage. It is the ontologist who chooses to apply the changes in the ontology, placing the new elements (concepts, relations), etc.

To reduce this frequent involvement of ontologist, (Ottens et al., 2009), (Sellami et al., 2012) propose an approach and a system based on an adaptive multi-agent system (AMAS) called DYNAMO MAS which automates three steps of stojanovic process (change capturing, representation phase, implementation of change). The role of ontologist in ontology evolution is fundamental. Thus, we must take into consideration its interventions. (Sellami et al., 2012) developed a system which allows to answer these several interactions with ontologist and one of its components is DYNAMO MAS which includes the ontology that will be proposed to the ontologist. Its inputs are candidate terms and lexical relations obtained from text analysis and an OWL ontology. It provides as output an ontology expressed in OWL. The ontologist gives his reaction by approving, moving, or rejecting the terms and concepts proposed by the AMAS.

Our contribution in this work is an extension of the approach of DYNAMO MAS (Sellami et al., 2012) by adding to the agents new behaviour to adapt and learn from ontologist feedback. His reaction is considered as perturbation by the DYNAMO MAS. As a feedback, it reorganizes the agents and produces another ontology proposition. This interactive process is repeated until a satisfactory state of the ontology is obtained. The originality of our system is to ensure the consistency of an ontology without the frequent involvement of the ontologist. It is an entire automatic adaptation instead of (Tissaoui et al., 2011) approach where the ontologist is involved in choosing the relevant scenario to apply the list of changes. In our approach, the ontologist only acts to give his desire for

Benomrane S., Sellami Z., Ben Ayed M. and Alimi A..
An Adaptive Multi-Agent System for Ontology Co-evolution.
DOI: 10.5220/0005257602160221
In *Proceedings of the International Conference on Agents and Artificial Intelligence* (ICAART-2015), pages 216-221
ISBN: 978-989-758-073-4

a proposal.

This article is organised as follows. Firstly, we present an overview of DYNAMO MAS and show our motivation to extend it. Secondly, we detail our extension in DYNAMO MAS. Finally, we conclude and plan some perspectives.

2 DYNAMO SYSTEM OVERVIEW

2.1 Ontology: A Self-organizing Multi-Agent System

DYNAMO system is an Adaptive Multi-Agent System (AMAS) based tool, supporting the co-construction and the evolution of a Terminological and Ontological Resources (TOR) to index documents and to allow semantic information retrieval. A TOR (called ontology in the rest of the paper) contains a set of domain concepts and a set of associated terms. Every term denotes at least one concept. DYNAMO system has a corpus of documents and even an ontology in OWL as input and generates an ontology as output.

The DYNAMO system includes three components: a corpus analyser, a MAS and a proposal manager (Fig. 1).

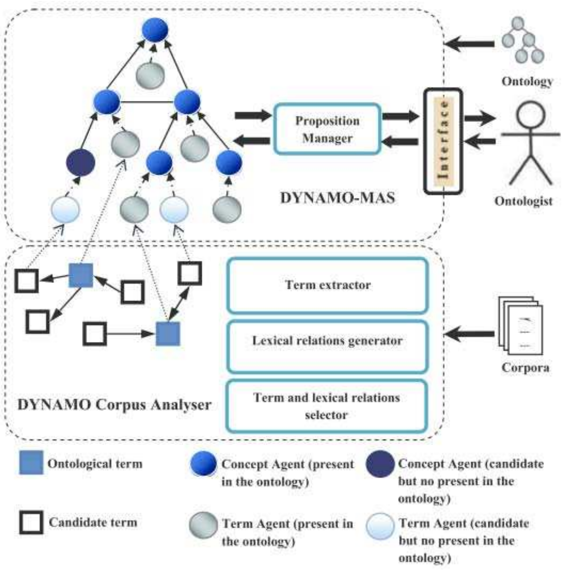

Figure 1: The general architecture of DYNAMO system (Sellami et al., 2013).

- The corpus analyzer: The goal of the corpus analyzer is to prepare inputs for the MAS. It runs the YaTeA term extractor (Sellami et al., 2013), a lexical relation generator, and a term and lexical relations selector. It outputs (Ti, Rel, Tj) triplets where Ti and Tj are candidate terms and/or terms (if the term belongs to the ontology) and Rel is a lexical relation label;
- The MAS: it includes the ontology that will be proposed to the ontologist. Its inputs are the triplets provided by the corpus analyzer and an OWL ontology. It provides as output an OWL ontology.
  Every agent has a relevance value which is the maximum confidence score of all its relations with its neighbors. During its life cycle, the agent computes its relevance value (Sellami et al., 2013). If it exceeds the proposal threshold (between 1 and 10), the agent proposes itself to be part of the ontology by sending a request to the proposal manager;
- The proposal manager: it enables the ontologist to both visualize the ontology and the MAS proposals and to interact with the MAS.

2.2 Agent Behaviour

DYNAMO MAS consists of two types of agents: (i) term agents represent candidates for the terminological part of the ontology and (ii) the concept agents represent candidates for the conceptual part of the ontology. The relations of the ontology as well as the lexical relations are part of an agent knowledge. A set of interactions between these agents enabling them to find the right position in the MAS organization.
Each agent acts in the MAS according to two types of behaviours:

- Nominal behaviour: it represents the basic algorithm allowing term agents and concept agents to move throw the MAS to retrieve their appropriate position (Sellami et al., 2011);
- cooperative behaviour: it represents the cooperative agents aptitude to resolve some exception situations called Non Cooperative Situations (NCS). Thanks to this behaviour, the AMAS ensures that it will converge for a solution. Thus, this cooperative process will put an end to a set of informations exchange to finally reach a stabilization (Sellami et al., 2011).

Obtained results (Sellami and Camps, 2012) show that the DYNAMO MAS is a very useful tool to the construction and the evolving of an ontology. However, these results present two key issues that should be solved so that the DYNAMO performs better:
(i) The term and concept agents are missing additional adaptive behaviour to react to ontologist actions. To fix it, we are supposed to enhance the behaviour of the

agents (nominal and cooperative) already developed and implemented in DYNAMO MAS (Sellami et al., 2013). We are adding these agents, adaptive skills, to detect its uselessness, to avoid the useless and wrong proposals and to propose others.
(ii) The time taken to automatically evolve an ontology is longer than necessary to manual maintenance. This is due to the response time during validation, deletion or modification of a proposal using PROMPTDIFF tool. We believe that optimizing the graphical interface between user and system, the time of the interaction with the ontologist will be considerably shortened.
The next section presents our proposed adaptive behaviour (an extension of DYNAMO MAS algorithm (Sellami et al., 2011)).

3 THE DYNAMO MAS EXTENSION

In this paper, we are focusing in the two latest components of DYNAMO system (the MAS and the proposal manager). First, the ontologist shows his reaction towards the evolved ontology (validates, refines, rejects, adds others terms or concepts . . .). These actions have an additional effects on the ontology elements (concepts, terms and/or relations). Secondly, the main part, MAS has to consider the actions of ontologist and learn from this feedback and propose a new ontology (new organization of MAS. This process is iterative until we achieve a consistent ontology (no more suggestions in the ≪proposal manager≫).

3.1 Types of Evolution Changes and their Consequences on the Ontology

To improve the different suggestions of DYNAMO MAS, agents behaviour are enhanced to react locally to a set of changes made by the ontologist (accept, reject, move, delete, create, split, merge, group. . .). Each change type can generate *additive changes*, by adding new elements (concepts, terms and/or relations) to the ontology without affecting the existing ones, and *substractive changes* by deleting some elements (Stojanovic, 2004).
Ontologist feedback can be a set of reactions towards the system proposals or some personal suggestions specific to him. Two kind of changes can be distinguished: elementary changes and composite changes. Stojanovic (Stojanovic, 2004) defines an elementary change as *an ontology change that modifies (adds or removes) only one entity of the ontology model* and a composite change as *a change of ontology that can be decomposed into several elementary changes*.
These changes require various ontology evolution strategies that depend on several criteria. The role of our AMAS is to determine automatically the appropriate strategy by adding to the agents new behaviour to adapt and learn from ontologist feedback. His reaction is considered as perturbation by the DYNAMO MAS. As a feedback, it reorganizes the agents and produces another ontology proposition. This interactive process is repeated until a satisfactory state of the ontology is obtained.
Our AMAS is based on the declarative approach of Stojanovic (Stojanovic, 2004). This approach does not specify in advance the possible strategies to resolve change operations. The ontologist specifies declaratively his desire (what) and the AMAS executes the appropriate evolution strategy (how) to meet the ontologist needs. The different types of evolution changes, considered in our approach, are listed and classified by elementary and composite changes in two tables as shown below:

Table 1: Term and Concept agents adaptive behaviour to ontologist elementary changes.

Term Agent adaptive behaviour	Concept agent adaptive behaviour
Adaptation to acceptance or rejection	Adaptation to acceptance
Adaptation to new term creation	Adaptation to rejection
Adaptation to removal	Adaptation to renaming
	Adaptation to removal
	Adaptation to sub-concept creation

Table 2: Term and Concept agents adaptive behaviour to ontologist composite changes.

Term agent adaptive behaviour	Concept agent adaptive behaviour
Adaptation to moving	Adaptation to moving
	Adaptation to merging
	Adaptation to renaming
	Adaptation to split
	Adaptation to grouping

3.2 Agents Adaptive Behaviour to Ontologist Feedback

DYNAMO MAS provides as output an ontology. The ontologist gives his reaction by applying one of the different evolution actions enumerated above. Our goal is to improve the algorithm of term and concept agent by adding new behaviour to adapt and learn from ontologist feedback. His reaction is considered as perturbation by the DYNAMO MAS. As a feedback, AMAS reorganizes the agents, improves the proposals already made and proposes others in order to generate a new ontology proposition. This interactive process is repeated until a satisfactory state of the ontology is obtained.

In total, we have 14 agent behaviours classified according to two criteria: Agent type (term or concept) and ontologist evolution action type (elementary actions or composite actions). In this paper, we present 4 examples of agent behaviours description that matches these criteria.

3.2.1 Term Agent Adaptive Behaviour to Ontologist Elementary Changes: Acceptance or a Rejection

To adapt to ontologist acceptance or rejection of a term, we enhance the term agent with the following behaviours: When a term agent is accepted or rejected by the ontologist, it informs its neighborhood, not yet validated, by sending a message. This message allows them to recompute their relevance value. For example, when the term agent *Problem* is accepted or rejected, it informs the term agent *Problem Failure* that has been accepted or rejected ❶ (Fig. 2). Then,

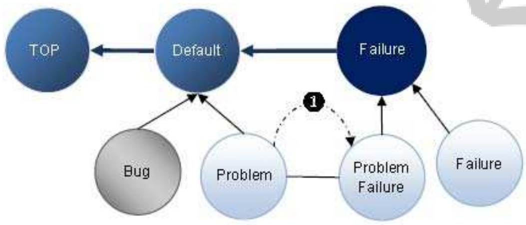

Figure 2: Term agent adaptation to an acceptance or a rejection.

it recalculates its relevance. If it exceeds the proposal threshold and it respects the Term Proposal Condition (TPC), it proposes itself to be part of the ontology.

TPC: a term agent can not be proposed if the concept that it denotes has not be proposed before it. When a concept agent is rejected then the term agents, that have not yet proposed, wait until the concept agent proposes itself again.

3.2.2 Term Agent Adaptive Behaviour to Composite Changes: Moving

When a term agent is moved by the ontologist, it informs its neighborhood, not yet validated, by sending a message. For example, if the term agent (Data Failure) is moved, then it sends a message to the concept agent (Data Failure) and the term agent (Problem Failure).

The term agent (Data Failure) moves to the concept agent (Failure) to denote it ❶ (Fig. 3). If the concept agent (Data Failure) is already proposed then it will be removed automatically.

If the term agents (Data Failure) and (Problem Failure) are related with a synonymy relation(Data Failure target)then the term agent (Problem Failure) computes its relevance value. If the moving of (Data Failure) to (Failure) increases the relevance of the term agent(Problem Failure) then (Problem Failure) moves to the concept agent (Failure) ❷.

If there is hyperonymy relation between (Data Failure) and (Problem Failure), then the term agent (Problem Failure) sends a request to the concept agent (Failure) asking for its parent concept agent ❷. The concept agent (Failure) processes the request and notifies the term agent (Problem Failure) by a message containing his parent concept agent (Default) ❸. (Problem Failure) sends a request to (Default) for establishing denotation link . (Default) accepts and sends a notification to (Problem Failure). Therefore, (Problem Failure) moves to (Default) to create a denotation link with it ❹. If the concept agent (Problem Failure) is already proposed then it will be removed automatically.

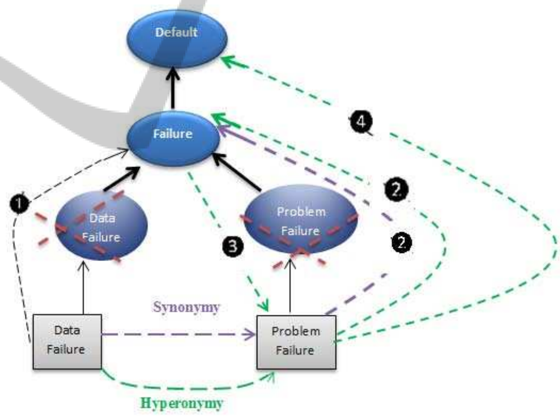

Figure 3: Term agent adaptation to moving.

3.2.3 Concept Agent Adaptive Behaviour to Elementary Changes: Removal

When the ontologist proposes to delete a concept agent, it informs its neighnorhood, not yet validated, by sending a message. This message allows them to recompute their relevance value.

For example, if the concept agent (Failute) has to be removed, it informs its neighnorhood, the term agent (Failure), its sub-concept agents (Data Failure) and (Problem Failure) and the concept agent (Exception) and it proposes a new parent concept agent (Default) ❶ (Fig. 4). The concept agents (Data Failure) and (Problem Failure) and the term agent (Failure) recompute their relevance value. If it exceeds the threshold

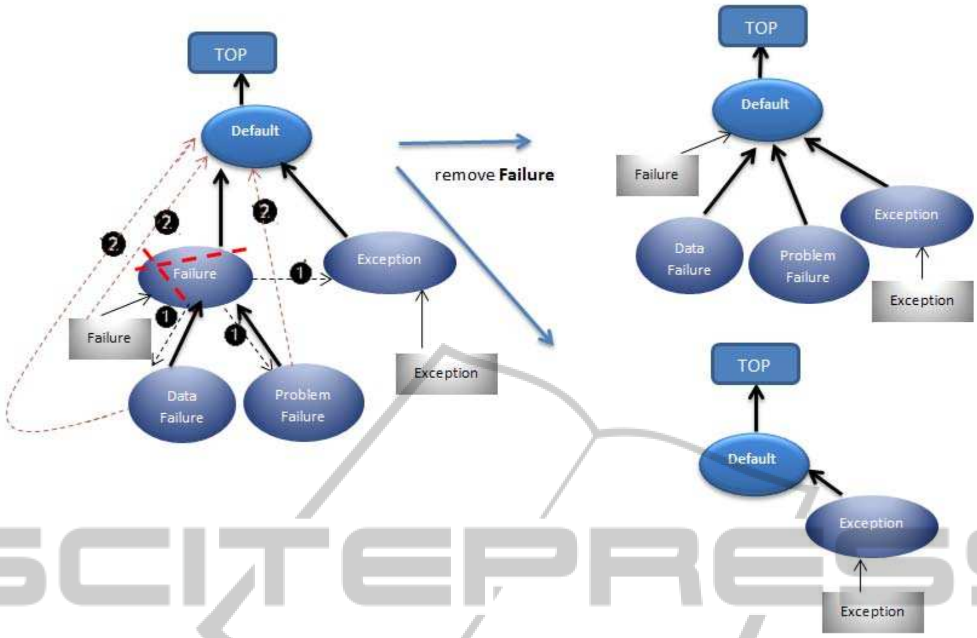

Figure 4: Concept agent adaptation to removal.

proposal, then, (Data Failure) and (Problem Failure) send a request for establishing an (is-a) relation with (Default) and the term agent (Failure) sends a denotation request to (Default) ❷. The latter accepts and notifies them. Therefore, (Data Failure), (Problem Failure) and the term agent (Failure) move to (Default). The concept agent (Failure) has no more a term to denotate it. Thus, it disappears from the AMAS. Else, the conecpt agent (Failute) will disappear with their sub-concept agents (they have no more parent concept agent) from the AMAS. If the relevance score of theconcept agent (Exception) exceeds the threshold proposal, then it can propose itself to the ontologist.

3.2.4 Concept Agent Adaptive Behaviour to Composite Changes: Merging

When the ontologist proposes to merge a concept agent with a validated one, the AMAS reacts and adapts to this action by running a set of operations.
For example, when the ontologist proposes to merge the concept agent (Exception) with the validated conecpt agent (Failure), (Exception) informs its sub-concepts agents(DB Exception) and (System Exception) and the term agent (Exception Error) by sending a message containing the new suggested parent concept agent (Failure) ❶ (Fig. 5). (DB Exception), (System Exception) send a request message to (Failure) for establishing an (is-a) relation and (Exception Error) sends a denotation request ❷. If their new relevance values(with the new parent concept) increase, then the concept agent (Failure) notifies them by sending a message of acceptation ❸. Therefore, (DB Exception), (System Exception) and (Exception Error) move to (Failure).
If (Exception) has no more sub-concepts or term agents then it disappears automatically from the AMAS else it will be removed, necessarily, by running the removal concept operation (detailed in section Adaptation to concept removal).

4 CONCLUSION

In this paper we present an automatic ontology evolution tool called DYNAMO MAS. This tool is based on an adaptive multi-agent system. First, DYNAMO MAS evolved an ontology in OWL from text. Secondly, it reacts to ontologist feedback and produces new ontology evolution proposal. We showed in Section 3 the different agents behaviour against elementary and composite changes. We noticed that the adaptive skills we added to term and concept agents allow them to detect the uselessness of some proposals, to avoid the useless and wrong ones and to propose others.
The originality of our work is that we exploit the ontologist feedback to improve the future propositions and reduce his frequent involvement.
To validate the efficiency of DYNAMO MAS extension, we are currently working in three tasks:

- Implementing the adaptive behaviour skills of our term and concept agents;

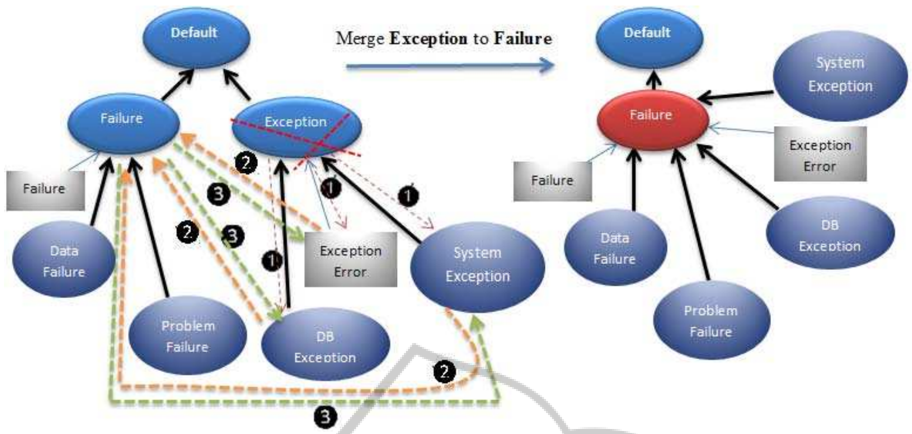

Figure 5: Concept agent adaptation to merging.

- Designing and developing a new DYNAMO MAS graphical interface between ontologist and MAS to enhance the time of the interaction and make it shortened;
- Testing the system with different ontologies in french and english languages.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

REFERENCES

Djedidi, R. and Aufaure, M.-A. (2010). Onto-evo a l an ontology evolution approach guided by pattern modeling and quality evaluation. In *Foundations of Information and Knowledge Systems*, pages 286–305. Springer.

Klein, M. C. A. (2004). *Change management for distributed ontologies*.

Luong, P. H. (2007). *Gestion de l'évolution d'un Web sémantique d'entreprise*. PhD thesis, École Nationale Supérieure des Mines de Paris.

Ottens, K., Hernandez, N., Gleizes, M.-P., and Aussenac-Gilles, N. (2009). A multi-agent system for dynamic ontologies. *Journal of Logic and Computation*, 19(5):831–858.

Sellami, Z., Aussenac-Gilles, N., Gleizes, M. P., and Camps, V. (2012). Dynamo, un outil de construction et d'évolution d'ontologies à partir de textes. *Technique et Science Informatiques*, 31(1):97–124.

Sellami, Z. and Camps, V. (2012). Evaluation of a multi-agent system for the evolving of domain ontologies from texts. In *Advances on Practical Applications of Agents and Multi-Agent Systems*, pages 169–179. Springer.

Sellami, Z., Camps, V., and Aussenac-Gilles, N. (2013). Dynamo-mas: a multi-agent system for ontology evolution from text. *Journal on Data Semantics*, 2(2-3):145–161.

Sellami, Z., Camps, V., Aussenac-Gilles, N., and Rougemaille, S. (2011). Ontology co-construction with an adaptive multi-agent system: Principles and case-study. In *Knowledge Discovery, Knowlege Engineering and Knowledge Management*, pages 237–248. Springer.

Stojanovic, L. (2004). Methods and tools for ontology evolution.

Tissaoui, A., Aussenac-Gilles, N., Laublet, P., and Hernandez, N. (2011). Evonto: évolution de ressource termino-ontologique pour lannotation sémantique. *Journées Francophones sur les Ontologies (JFO 2011)*, pages 22–23.