DYNAMIC ONTOLOGY CO-CONSTRUCTION BASED ON ADAPTIVE MULTI-AGENT TECHNOLOGY

Sellami Zied, Gleizes Marie-Pierre, Aussenac-Gilles Nathalie
IRIT (Institut de Recherche en Informatique de Toulouse), Toulouse University, 118 Route de Narbonne, F-31062 Toulouse cedex 9, France
surname.name@irit.fr

Rougemaille Sylvain
UPETEC (Emergence Technologies for Unsolved Problems), 10 avenue de l'europe, 31520 Ramonville, France
sylvain.rougemaille@upetec.fr

Keywords: Ontology engineering, Multi-Agent System, Knowledge acquisition

Abstract: Ontologies have become an important means for structuring knowledge and defining semantic information retrieval systems. Ontology engineering requires a significant effort, and recent researches show that human language technologies are useful means to acquire or update ontologies from text. In this paper we present DYNAMO, a tool based on a Multi-Agent System, which aims at assisting ontologists during the ontology building and evolution processes. This work is carried out in the context of the DYNAMO project. The main novelty of the agent system is to take advantage of text extracted terms and lexical relations together with some quantitative features of the corpus to guide the agents when self-organizing. We exhibit the first experiment of ontology building that shows promising results, and helps us to identify key issues to be solved to the DYNAMO system behavior and the resulting ontology.

1 INTRODUCTION

One way to provide efficient search on a document retrieval system is to explicitly state the meaning of document contents. On-going research in this area tries to address the problem by tagging and indexing the contents of documents thanks to an organised knowledge representation called ontology.

Ontologies are often used to represent a specification of domain knowledge by providing a consensual agreement on the semantics of domain concepts, or an agreement on the concepts required for a specific knowledge intensive application. An ontology also defines rich relationships between concepts. It allows members of a community of interest to establish a shared formal vocabulary. In short, ontologies are defined as a formal specification of a shared conceptualisation (Gruber, 1993) where formal implies that the ontology should be machine-readable and shared that it is accepted by a human group or community. Further, it is restricted to the concepts and relations that are relevant for a particular task or application.

Typically, ontologies are composed of a hierarchy of concepts the meaning of which is expressed thanks to their relationships and to axioms or rules that may constrain the relations or that define new concepts as formulas. Concepts may be labelled with terms that are their linguistic realisations or linguistic clues of their meaning.

Originally, ontologies were defined as rigid structures that are supposed to be stable over time. Nevertheless, ontologies may need to evolve because domain knowledge changes, users' needs may be different or because the ontology could be used in a new context or even reused in a new application (Haase and Sure, 2004). Ontology maintenance may result difficult especially if their structural semantics is complex, defining hundreds of concepts and relations.

Ontology engineering is a costly and complex task (Maedche, 2002). In the last ten years, ontology engineering from text has emerged as a promising way to save time and to gain efficiency for building or evolving ontologies (Buitelaar et al., 2005). However, texts do not cover all the required information to build a relevant domain model, and human interpretation and validation are required at several stages in this process. So ontology engineering remains a particularly complex task when it comes to the extraction or observation of both terminological and ontological representations from a specific document corpus.

Our motivation is to propose a tool facilitating ontology maintenance and evolution by the ontologist. The principle is to provide a system that automatically proposes solutions to be discussed and evaluated. This system learns from the user's feedback. It can be seen as a virtual ontologist that helps the "real one" to carry out ontology learning and evolution from text. We call this process a co-construction or co-evolution.

In this article, we propose DYNAMO, a Multi-Agent System that supports ontology co-construction and evolution. After an overview of the DYNAMO context (section 2), we detail the principles of this MAS (section 3). In Section 5 we exhibit preliminary results obtained when building an ontology with DYNAMO. These results are discussed in the next section. Section 7 presents the improvements we plan to bring to our work as well as our perspectives.

2 DYNAMIC ONTOLOGY CO-CONSTRUCTION

2.1 Context

The study of ontology evolution is part of the DYNAMO[1] (DYNAMic Ontology for information retrieval) ANR[2] (Agence Nationale de la Recherche) funded research project. DYNAMO addresses the improvement of semantic information retrieval driven by user satisfaction in a dynamic context. One of the project originality is to take into account the potential dynamics of the searched document collection, of the domain knowledge as well as the evolution of users' needs.

The DYNAMO project aims at proposing a methodological approach and a set of tools that allow the definition and the maintenance of ontological resources from a set of unstructured documents. These resources are used to facilitate information retrieval within the corpus by means of a semantic indexing.

Several project partners propose domain specific document collections. ACTIA[3] provides documents covering the area of automobile diagnosis (automobile components, symptom, engine failure, etc.) and written in French, while ARTAL[4] corpus consists in software bug reports written in English, and the partners in charge of the ARKEOTEK[5] project are concerned by archaeological scientific research papers structured as a set of rules. Thus, one point of importance in DYNAMO tools, is the handling of a variety of heterogeneous document collections among the projects (as for language support both in French and English) for the co-evolution of ontological resources.

2.2 The Ontology Model

In DYNAMO, the ontology and its lexical component form what we call a Terminological and Ontological Resource (TOR). Such a resource is represented using the OWL[6]-based TOR model proposed in (Reymonet et al., 2007). This model recently evolved to become a meta-model, where concepts and terms are two meta-classes adapted from owl:class. In this TOR, model ontological elements (concepts) are related to their linguistic manifestations in documents (terms): a term "denotate" at least one concept. This models form the core of the DYNAMO project as longs as term instances (which represent term occurrences), concept instances and relations between instances are used to represent document annotations.

The problem addressed in this paper is how a multi-agent system can be used to build and update a TOR represented with this meta-model and using documents as information sources.

2.3 The Adaptive Multi-Agent System (AMAS) Theory

Because of their local computation and openness, MAS are known to be particularly well fitted to dynamic and complex problems. The design of an ontology from the analysis of a corpus is an obviously complex task (as we discussed in the introduction).

The AMAS Theory (Capera et al., 2003) emphasises on the cooperation between agents to achieve global adequacy by the way of self-organisation. Each agent in the system tries to maintain a cooperative state, more precisely, it tries to avoid and repair harmful situations (Non Cooperative Situations). According to the AMAS principles, the agent cooperative behaviour ensures that, during times, the function realised by the system is always adapted to the problem (functional adequacy). The main idea of the work presented here is to take advantage of the AMAS properties to propose an ontology co-construction system that uses as information sources the documents as well as the ontologist interactions.

[1] http://www.irit.fr/DYNAMO/

[2] http://www.agence-nationale-recherche.fr/

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

[4] http://www.ARTAL.fr

[5] http://www.ARKEOTEK.org

[6] Web Ontology Language http://www.w3.org/2004/OWL/

As a result, the DYNAMO MAS proposes some modifications to the ontologist because it evaluates that these modifications improve the ontology. The system also benefits from the ontologist answers to its proposals (basically acceptance or rejection), it allows to strengthen or weaken the confidence in the position of the involved agents.

3 PRINCIPLES

DYNAMO is a tool, based on an Adaptive Multi-Agent System (AMAS) presented in Section 2.3, enabling the co-construction and the maintenance of an ontology starting from a textual corpus and resulting an OWL file. DYNAMO is a semi-automatic tool because the ontologist has to validate, refine or modify the organisation of concepts, terms and relations between concepts until it reaches a satisfying state. Figure 1 gives an overview of the DYNAMO system components : *DYNAMO Corpus Analyzer* and *DYNAMO MAS*.

The *DYNAMO Corpus Analyzer* prepares the input of the *DYNAMO MAS*. It contains the *Corpus*, the *Pattern Base*, the *Candidate Term Base* and a set of NLP tools. Those tools process the lexical relations extraction mechanism whose result is used to determine potential semantic relations between candidate terms.

The DYNAMO MAS is composed of two agent types: *TermAgent* and *ConceptAgent* which are detailed further in Section 3.3. Thanks to the extracted relations these agents try to self-organise in order to find their proper location in the TOR hierarchy.

3.1 Syntactic Patterns

Many approaches for ontology learning from text are based on Natural Language Processing (NLP) techniques. We can quote two main groups: on the one hand, statistical approaches (Harris, 1968), like clustering, are interested in finding a semantic interpretation to several kinds of term co-occurrences in corpora; on the other hand, approaches based on linguistics rely on a more or less fine-grained linguistic description of the language used in text to derive an interpretation at the semantic level. Recent ontology learning processes combine both approaches (Cimiano, 2006).

For instance, lexico-syntactic patterns can be used either for concept or semantic relation extraction, but what they actually identify in text are terms or lexical relations (Hearst, 1992). The extraction process then includes pattern adaptation to the corpus to be parsed, lexical relations extraction on each document, phrase interpretation and finally term and relation extraction (Barrière and Akakpo, 2006). An extra step would be to define concepts and semantic relations from those items. Systems like Text-to-Onto (Cimiano and Völker, 2005) or OntoLearn (Velardi et al., 2005) propose a fully automatic run from pattern-matching to ontology learning, while systems like Prométhée (Morin, 1999) and Caméléon (Chagnoux et al., 2008) support a supervised process when the ontologist may validate or modify the concepts and relations proposed by the analysis tool.

In keeping with results established by V. Malaisé (Malaise, 2005) we have experimentally observed that a statistical processing is not very effective on small corpora with little redundancy, which is the case for the three DYNAMO specific applications. Not only the corpora have a relatively modest size (ACTIA corpus: 46000 words, ARTAL's corpus: 13000 words, ARKEOTEK corpus: 106000 words), but each document is of very short length and deals with a specific subject. For all these reasons, we adopted a pattern-based approach to obtain relevant information on terms and their relationships, and then to define concepts and semantic relations from these evidences.

3.2 Semantic Relations

In DYNAMO, we are interested in four types of lexical relations:

1. Hypernymy expresses a generic-specific relation between terms. This may lead to define a class-subclass (*is_a*) relation between the concepts denoted by these terms.

2. Meronymy means a parthood relation between terms, which may lead to define a *part_of* semantic relation between concepts, or an *ingredient_of* relation or domain-specific adaptations of parthood like *has_members* in biology for instance.

3. Synonymy relates semantically close terms that should denote the same concept.

4. Functional relations: which are any other kind of lexical relations that will lead to a specific set of semantic relations, either general ones like *causes, leads_to, ...* or task specific relations like *has_fault*, *is_an_evidence_for* or domain specific relations like *has_skills* in archaeology.

In our system, linguistics manifestation of semantic relations are used by agents as clues for self-organisation. We call them triggers.

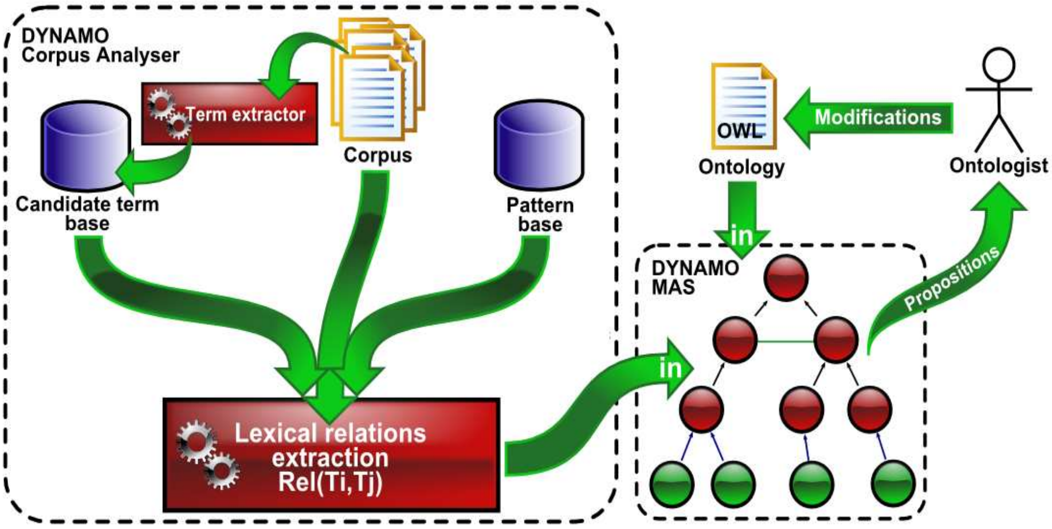

Figure 1: Relations between the *DYNAMO Corpus Analyzer*, the *DYNAMO MAS* and the ontologist

3.3 Agents Behaviour

The Multi-Agent System is constituted of two different types of agent: one representing the terminological part of the TOR (*TermAgent*) and the other, the conceptual part (*ConceptAgent*). Considering an ontology as the MAS organisation, the aim of our system is to reach a stable organisation of terms and concepts according to the semantic relations extracted from the considered corpus. This is mainly achieved through a cooperative self-organisation process between the agents encapsulating all these elements.

3.3.1 TermAgent Behaviour

TermAgents represent terms that have been extracted from the corpus. Each extracted term is related to one or many other terms by either syntactic relations or lexical relations, which are linguistic manifestations of semantic relations. These relations have been detected thanks to the lexical relations extraction mechanism (essentially using specific triggers).

The system is initialised by creating one *TermAgent* from each extracted term. The agent behaviour consists then in processing all the extracted relations that connect it with other terms. Each relation has a confidence degree which is computed from the frequency of occurrences of the corresponding pattern. Using this confidence degree *TermAgents* are related with each other (see 3.2).

Initially the MAS is only composed of *TermAgents* which are linked by these valued relations. Each *TermAgent* takes into account the most important relation (which has the greatest confidence value) according to its type:

1. Synonymy relation between two *TermAgents* leads to the creation of a *ConceptAgent* linked to the corresponding *TermAgents*.
2. Hyperonymy relation between two *TermAgents* causes the creation of two *ConceptAgents* linked to the corresponding *TermAgents*. The two new *ConceptAgents* are related by an *is_a* relation.
3. Meronymy relation between two *TermAgents* implies the creation of two *ConceptAgents* linked to the corresponding *TermAgents*. The new *ConceptAgents* are associated with a *part_of* relation.
4. Functional relation between two *TermAgents* leads to the creation of two *ConceptAgent* linked to their corresponding *TermAgents*. The two new *ConceptAgents* are then related by this relation.

TermAgents send a request to create *ConceptAgents* and link with it. Additionally, when a *ConceptAgent* previously exists, it is not created a second time.

3.3.2 ConceptAgent Behaviour

This agent type represents concepts that have been created by *TermAgents*. *ConceptAgents* behaviour leads to the optimisation of the ontology by treating a set of Non Cooperative Situations (NCS) derived from specific data about the three relations type (hyperonymy, meronymy and functional) as well as their link to terminological data (*TermAgent* denotation). An example of these situations can be found in the election of a concept label. Typically, each *ConceptAgent* has to choose among its related terms the one that is the more representative to become its label. To do so, a *ConceptAgent* selects the denotation relation on which it is the most confident, and proposes to the designated *TermAgent* to become its label. However, conflicts may appear in this process. As a single *TermAgent* could denote several *ConceptAgents* it may receive several label requests. This situation is quoted as an NCS (a conflict one), it is detected by a *TermAgent* and should be treated by a *ConceptAgent*. Several solutions could be adopted at this stage:

- If there is only one common *TermAgent* (the label) linked to the concerned *ConceptAgents*, they should be merged.
- If there are several other *TermAgents* linked to the *ConceptAgents*, *ConceptAgents* have to chose another label in their *TermAgents* pool. The *TermAgent* that detects the situation kept as a label by one of the *ConceptAgents* depending on its similarity to other designated *TermAgents*.

Between these two cases a mid-term solution have to be found, by considering the number of *TermAgents* linked to each *ConceptAgents*, the similarity between these *TermAgents*, the relation held by the denoted *ConceptAgent*, etc. This is achieved through cooperation and thanks to the ontologist actions. The cooperation at the system level is the purpose of the following section.

3.3.3 Collective Behaviour

The *DYNAMO MAS* is a real time system that uses a corpus that implies the creation of several hundreds of *TermAgents* and *ConceptAgents*. We need to be sure that the system converges and outputs at least a solution. This convergence is guaranteed by the AMAS theory. In short, because agents are implemented in such a way that they can be considered as cooperative, their cooperation ensures that the whole set will stabilize after a large set of iterations for information exchange. In fact, the collective process stops when each agent reaches a local equilibrium. This equilibrium occurs when its remaining NCS levels are lower than the NCS levels of its neighbourhood (agents that are related to it). For example, let us consider a given *TermAgent* looking at its relations with other *TermAgents* which are not currently in the TOR:

- If a neighbour agrees to take their relationship into account, the MAS changes the ontology by adding the new relationship. This modification is proposed later to the ontologist for agreement. If the ontologist disagrees the MAS stores this information to avoid the same request one more time.
- If related neighbours disagrees, about this modification because of contradictions with other more critical situations in its own neighbourhood, no change occurs.

Semantic Relation	Number of Triggers
SYNONYMY	12
HYPERONYMY	21
MERONYMY	23
FUNCTIONAL	16

Table 1: Number of triggers for relation extraction

Furthermore, we considered a minimum confidence threshold in the algorithm, in order to dismiss a large number of non-significant semantic relations extracted from the corpus. The confidence degree of any relation could evolve when new documents are analysed. By this means, relations that were set apart could be later taken into account providing that their confidence degree reaches the threshold. This simple rule also avoids processing relations considered as analysis noise.

The collective solving process of agents is also very efficient for algorithmic reasons:

- the AMAS algorithm assumes a monotonic decreasing of NCS level: typically three or four agent activations are sufficient to obtain a local equilibrium;
- when new information arrives in the MAS (coming from new corpus analysis or the ontologist) only the considered agents work. Thus the perturbation process is very limited inside the MAS.

4 EXPERIMENTS

We experimented our system using the ARTAL corpus defined in the DYNAMO project. The objective of the experiment is to analyse the results of the system and to improve *TermAgent* and *ConceptAgent* behaviour to obtain an ontology.

The pattern base was fed with some of the triggers defined in the TerminoWeb project (Barrière and Akakpo, 2006). These triggers are used to extract relations between terms from the corpus (see 3.1). The following list presents some of the chosen triggers:

- for hyperonymy relations: *such as, and other, including, especially*;
- for meronymy relations: *is a part of, elements of, components of*;
- for synonymy relation: *another term for, also called, also known as, synonym*.

Semantic Relation	Matched relation number
SYNONYMY	2
HYPERONYMY	68
MERONYMY	36
FUNCTIONAL	370

Table 2: Number of matched relations in the ARTAL corpus

The precise numbers of triggers used for each relation type are presented in Table 1. We also used specific triggers for functional relations such as: *when, if, at, on, before, after*.

Due to the small number of triggers, we estimate potential semantic relations between terms by the following rule: if it is possible to find two terms sharing the same contexts in the corpus (i.e. often situated among same words), then these terms could be considered as potential semantically related. The confidence of this relation is calculated using the Jacquard Formula based on the shared context cardinality. For example, the *DYNAMO Corpus Analyser* considered “Web”, “Applet” and “Service” as potential hypernym of the concept “Application” because they share same context.

Finally, we filled the *Candidate terms base* with the terminological part of the ARTAL TOR which contains 692 terms.

5 RESULTS AND ANALYSIS

For the experiment, we use a strict matching of terms on the corpus, we do not use any of their orthographic variations. This fact explains the relatively small number of matched relations and facilitates the readability of MAS results. Using *DYNAMO Corpus Analyser*, we have extracted 476 relations between 503 terms. Table 2 gives the number of relations detected using the triggers presented in the previous section.

The small amount of synonymy, hyperonymy and meronymy matched relations is due to the use of generic triggers that are obviously not fit to the corpus. On the opposite, we use specific triggers to extract functional relations and this is why it shows the best result. This point highlights the main feature of pattern based approaches which is the strong dependencies of the results on the choice of patterns.

Thanks to the used rule we have extracted 650 potential semantic relations between all terms. The MAS input is formed by 657 candidate terms and 1124 instance of relations.

Firstly, all terms are *agentified* and every agent searches to be related with a *ConceptAgent*. In Figure 3, links without label represent denotation relations between *TermAgents* and *ConceptAgents*. In the ontology model, each term can denote several concepts. Each labelled link represents the functional relation discovered between terms. For example, *Action* and *Activation* concepts are related by a link labelled *On*.

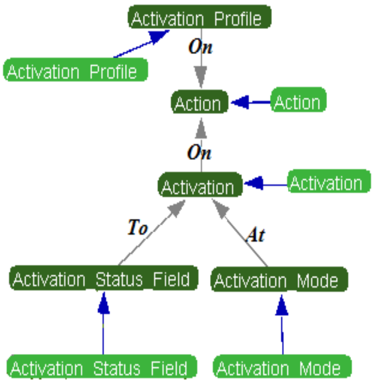

Figure 2: Functional relations established between *ConceptAgents*

In the considered domain (software bug reports) the presented subgraph signifies that an *Action* triggers two kinds of activation. The *Activation* possessed several modes which can be selected and modifies the status of the associated component (*Activation_status_field*). This representation is close to the one proposed by the ontologist, but expresses also some new functional relations.

Figure 3 presents a result obtained with *TermAgent* reasoning only with Jaccard Formula. Five *TermAgents* are related to the “Application” *ConceptAgent* who is related to the top concept. This first graph represent a potential hypernym relation between Agents. In the second graph five *TermAgents* (Impossible_to_send, Not_activated, Have_the_same_name, can_not_be_change, Https_port_number) are related to the “Https_port_number” *ConceptAgent*. Relation established between this *ConceptAgent* and the forth first *TermAgent* represent a potential functionnal relation (affected_by). In the considered domain (software bug reports), the relation means that http port can have several bugs like a not activation problem.

To improve our results, we need to define more precise patterns by using regular expressions rather than triggers which only detect possible relations between words situated on their both sides. We are also investigating on the use of external information re-

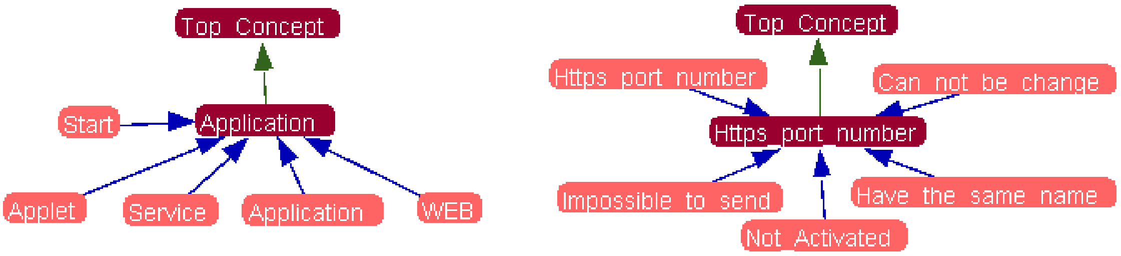

Figure 3: Example of semantic relation between *TermAgents* and *ConceptAgents* identified with jaccard Formula

sources (dictionary, generic ontology as WordNet[7], other domain related corpus, etc.) to deal with the limitation of corpus based approaches.

To improve agents behavior, we can also combine the Pattern-base approach with a statistical approach by using the Jaccard Formula. For example, a TermAgent can improve their trust in a relation if the same relation is also detected using Jaccard Formula.

As it has been expressed in Section 3.3, we do not have completely specified the *ConceptAgent* behaviour, this prototyping phase is part of the ADELFE methodology (Bernon et al., 2005). The aim of this specific task is to obtain the first draft of the system that allows to highlight more efficiently the NCS. Thanks to our first experiments, we have been able to quote some NCS as, for instance, the label conflict described in Section 3.3.2.

6 RELATED WORKS

6.1 The Earlier DYNAMO Prototype

The objective of DYNAMO is to facilitate ontology engineering from text thanks to a combination of Natural Language Processing and a cooperative Multi-Agent System. Our research is inspired from DYNAMO first prototype (Ottens et al., 2008) that used a statistical approach to build up a taxonomy from large text corpora. In this prototype, agents implement a distributed clustering algorithm that identify term clusters. These clusters lead to the definition of concepts as well as their organisation into a hierarchy. Each agent represents a candidate term extracted from the corpus and estimates its similarity with others thanks to statistical features. Several evaluation tests conducted with this DYNAMO first prototype proved its ability to build the kernel of a domain ontology from a textual corpus.

6.2 Ontology Engineering from Text in Dynamic Environments

Two on-going major IST European projects, SEKT[8] and NEON [9] aim at similar goals with a more ambitious scope. Both of them are building up toolkits that should give access to a panel of technologies, including several Human Language Technologies among which NLP plays a major role. SEKT and NEON want to advance the state of the art in using ontologies for large-scale semantic applications in distributed organisations and dynamic environments. Particularly, they aim at improving the capability to handle multiple networked ontologies that exist in a particular context, they are created collaboratively, and might be highly dynamic and constantly evolving. Human Language and Ontology Technologies are combined to produce semi-automatic tools for the creation of ontologies, the population of those ontologies with metadata, and the maintenance and evolution of the ontologies and associated metadata. Although agents technology is not used at all in these projects, their scope is very similar to the one of DYNAMO.

The ambition of SEKT is to offer this variety of technologies to develop not only ontologies and annotations, but full knowledge management or knowledge intensive applications. Argumentation among ontology authors who locally update an ontology is considered as a key stage of the evolution process of shared ontologies. The NEON project highlights the role of NLP when updating ontologies in dynamic environments together with their related semantic metadata. NEON toolkit offers a tool suite that extends OntoStudio baseline and connects it with GATE. GATE used to propose Protégé as a plug-in for ontology development from text analysis. The new GATE version includes a module that manages its own ontology representation, a plug-in for ontology population and text annotation.

[7] http://wordnet.princeton.edu/

[8] http://www.sekt-project.com/

[9] http://www.neon-project.org/web-content/

7 CONCLUSION

DYNAMO is a Multi-Agent System allowing the co-construction and evolution of ontologies from text. DYNAMO MAS uses result from lexical relations extraction mechanism to construct ontologies. We shown in Section 5 that the system is able to create an ontology draft containing both ontological and terminological elements and enriches several dimensions of the previous prototype (see Section 6):

1. It is able to deal with richer linguistic information as long as agents take into account lexical relations found by matching patterns on texts.
2. The result is much richer: DYNAMO builds up a TOR which includes a hierarchy of concepts with their related terms, and labelled semantic relations between concepts. A set of terms denotes each particular concept, which is useful for the document annotation activity.
3. The current DYNAMO system is able to deal either with French or English language text, whereas the first prototype was previously limited to French language.

According to the project schedule we need to improve the software during the next year. To do so we plan to:

- introduce the cooperative behaviour of *ConceptAgents* (specification of NCS and their treatment);
- provide an adaptive patterns learning process based on the AMAS theory;
- provide specific interfaces to enable ontologists collaboration;
- apply the *DYNAMO MAS* to all the project domains (archeology, car diagnosis, software bug reports).

REFERENCES

Barrière, C. and Akakpo, A. (2006). Terminoweb: A software environment for term study in rich contexts. In *Proceedings of International Conference on Terminology, Standardization and Technology Transfer*, pages 103–113, Beijing. Encyclopedia of China Publishing House.

Bernon, C., Camps, V., Gleizes, M.-P., and Picard, G. (2005). Engineering Adaptive Multi-Agent Systems: The ADELFE Methodology . In Henderson-Sellers, B. and Giorgini, P., editors, *Agent-Oriented Methodologies*, volume ISBN1-59140-581-5, pages 172–202. Idea Group Pub, NY, USA.

Buitelaar, P., Cimiano, P., and Magnini, B. (2005). *Ontology Learning From Text: Methods, Evaluation and Applications*. IOS Press.

Capera, D., Georgé, J.-P., Gleizes, M.-P., and Glize, P. (2003). The AMAS Theory for Complex Problem Solving Based on Self-organizing Cooperative Agents . In *TAPOCS 2003 at WETICE 2003, Linz, Austria, 09/06/03-11/06/03*. IEEE CS.

Chagnoux, M., Hernandez, N., and Aussenac-Gilles, N. (2008). An interactive pattern based approach for extracting non-taxonomic relations from texts. In Buitelaar, P., Cimiano, P., Paliouras, G., and Spilliopoulou, M., editors, *Workshop on Ontology Learning and Population (associated to ECAI 2008) (OLP), Patras (Greece), 22/07/08*, pages 1–6.

Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer.

Cimiano, P. and Völker, J. (2005). Text2onto - a framework for ontology learning and data-driven change discovery. In Montoyo, A., Munoz, R., and Metais, E., editors, *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238, Alicante, Spain. Springer.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–221.

Haase, P. and Sure, Y. (2004). D3.1.1.b state-of-the-art on ontology evolution. Technical report, Institute AIFB, University of Karlsruhe.

Harris, Z. S. (1968). *Mathematical Structures of Language*. Wiley, New York.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545.

Maedche, A. (2002). *Ontology learning for the Semantic Web*, volume 665. Kluwer Academic Publisher.

Malaise, V. (2005). *Méthodologie linguistique et terminologique pour la structuration d'ontologies différentielles à partir de corpus textuels*. PhD thesis, Paris 7 Denis Diderot University.

Morin, E. (1999). Using lexico-syntactic patterns to extract semantic relations between terms from technical corpus. In *5th International Congress on Terminology and Knowledge Engineering (TKE)*, pages 268–278, Innsbruck, Austria. TermNet.

Ottens, K., Hernandez, N., Gleizes, M.-P., and Aussenac-Gilles, N. (2008). A Multi-Agent System for Dynamic Ontologies. *Journal of Logic and Computation, Special Issue on Ontology Dynamics*, 19:1–28.

Reymonet, A., Thomas, J., and Aussenac-Gilles, N. (2007). Modelling ontological and terminological resources in OWL DL. In Buitelaar, P., Choi, K.-S., Gangemi, A., and Huang, C.-R., editors, *OntoLex07 - From Text to Knowledge: The Lexicon/Ontology Interface Workshop at ISWC07 6th International Semantic Web Conference, Busan (South Korea), 11/11/07*.

Velardi, P., Navigli, R., Cucchiarelli, A., and Neri, F. (2005). *Evaluation of OntoLearn, a Methodology for Automatic Learning of Domain Ontologies. In Buitelaar, P., O. Cimiano & B. Magnini (eds.)*. IOS Press, Amsterdam.