

Agent and Ontology based Information Gathering on Restricted Web Domains with AGATHE

Bernard Espinasse
LSIS UMR CNRS 6168

Domaine Universitaire de St Jerôme, F-13997, Marseille Cedex 20, France
bernard.espinasse@univ-cezanne.fr

Sébastien Fournier
LSIS UMR CNRS 6168

Domaine Universitaire de St Jerôme, F-13997, Marseille Cedex 20, France
sebastien.fournier@lsis.org

Fred Freitas
Universidade Federal de Pernambuco, CIn-UFPE, Centro de Informática, Caixa Postal 7851 50372-970, Brazil
fred@cin.ufpe.br

ABSTRACT

Due to Web size and diversity of information, relevant information gathering on the Web is a very complex task. The main problem with most information retrieval approaches is neglecting the context of the pages, mainly because search engines are based on keyword indexing. Considering restricted domain, the policy of taking into account context may lead to more relevant information gathering in this paper, a specific cooperative information gathering approach based on the use of software agents and ontologies is proposed. To implement this approach, a generic software architecture, named AGATHE system, based on early prototype, the MASTER-Web system, permitting development of specific restricted-domain information gathering systems is presented in detail, with a focus on the extraction subsystem.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: *Information filtering*
I.2.11 [Distributed Artificial Intelligence]: *Multiagent systems*

General Terms

Algorithms.

Keywords

Multiagents, information agents, agent-oriented software engineering, cooperative information gathering, information extraction, classification

1. INTRODUCTION

Without taking into account the explicit context in which a search is made, the majority of current information retrieval (IR) approaches let any organized form of Web information processing escape, for example of the specific regroupings or "clusters" of information (academia, tourism, ...). Considering such clusters, grouping classes of pages with their information (for example the whole of pages displaying the data of a stock exchange, the whole of pages on researchers...), would allow the recognition and the classification of pages dispersed on the Web, relative to a restricted field, and consequently to facilitate context processing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

Copyright 2008 ACM 978-1-59593-753-7/08/0003...\$5.00.

Taking context into account, better data processing is then possible. It is the case for information extraction of the majority of from pages about the same topic (for example value of the dollar, subjects of interest of a researcher...).

Another advantage is to make possible for the users to carry out queries combining criteria relative to various classes of pages, thus allowing complex queries (the search of the papers published in a certain series of conferences for example). Thus the realization of sophisticated applications for collecting information on the Web for specific fields is possible. With the "tourism" cluster, for example, it would be possible to realize a system that takes into account hotels, passage-tickets, and cultural events related to a scientific event.

This search concerns an approach of the collection of information on Web restricted domain taking into account the context of search. It builds upon the MASTER-Web architecture [5], and the major enhancements over MASTER-Web are the many new constitutive subsystems that are designed in order to assure flexibility, extensibility, scalability and reusability. The basis of these approaches lie in the use of ontologies that make possible to define restricted field and context, and the cooperative way using software agents, in particular to perform information extraction.

First of all, this paper introduces the interest of using software agents and ontologies to develop intelligent gathering systems. Then, section 3 introduces AGATHE system, generic software architecture for development of intelligent gathering systems on the Web on one or more restricted domain. AGATHE architecture is presented in a general way, then each of its constitutive subsystems: Search, Extraction and Front Office subsystems. In section 4 the central Extraction subsystem of AGATHE is presented in detail. Section 5 presents some implementation details related to the prototype in progress and some first results are given. Finally we conclude with some perspectives for research at short, middle and long terms.

2. INFORMATION GATHERING BASED ON AGENTS AND ONTOLOGIES

In this section, the use of software agents and ontologies for information gathering on the Web is argued.

2.1 Software agents and co-operative information gathering

The software agents come from the multiagent systems (MAS), which constitute one of the three principal branches of the

distributed artificial intelligence (DAI), two others being respectively the resolution of distributed problems and parallel artificial intelligence [13]. The software agents inherit consequently the properties and potentialities usually associated with the field of the DAI. For example, they inherit the modularity, the speed of execution (due to parallelism) and the reliability (thanks to the redundancy) of the systems of this field. Other properties coming from the traditional AI also come to be added in particular allowing a handling of knowledge, an easy maintenance, a re-use and independence of the platforms [8].

The MAS and software agents were first of all used in order to solve intrinsically distributed problems and to model and simulate complex phenomena (Agents Based Modelling and Simulation). Henceforth they are also used in order to develop systems of information retrieval and information classification. Such systems give the possibility of designing robust and adaptive tools. The FIPA (Foundation For Physical Agents) proposes many standards and recommendations facilitating the use of the paradigm agent in real applications. Lastly, several methodologies were also proposed making it possible to support the design and implementation phases contributing to the constitution of an agent oriented software engineering (AOSE).

The concept of "Cooperative Information Gathering" (CIG), suggested by Oates [14], is based on distributed problem-solving paradigm and its attendant body of research in MAS and DAI. The CIG involves concurrent, asynchronous discovery and composition of information spread across a network of information servers. The distributed resolution of problem is then a means for the agents of discovering relevant clusters of information. Other research tasks [1] recommend the use of agents for the collection of information: the databases of a large numerical library were grouped in hierarchical classes, each class having its own agent with explicit knowledge about it. These agents build the research plans, which improve the effectiveness in the process of search. The use of such tool requires a correct pairing of Web pages with these hierarchical classes and to extract information from them, in order to feed databases.

2.2 Cooperative information gathering and ontologies

In information gathering on the Web, the combination of the multi-agents approach with declarative knowledge, leading to the ontologies use, is relevant for the intelligent information gathering systems development.

First of all, this combination is justified because of the traditional advantages of the declarative solutions over those that are procedural. The declarative solutions of knowledge engineering provide a closer integration of the ontological approach with a more direct translation of the domain knowledge. The tasks of extraction and classification on the Web, which imply unstructured or semi structured data, require frequent changes of their solutions. With declarative knowledge, such changes can be easily taken into account, without recompiling of code or stop of execution. What constitutes a notable advantage of extensibility?

The great expressivity of the declarative knowledge representation is also a major advantage in collection of information on the Web. In most of the possibilities of inferences, when concepts implied in these tasks (for example cluster entities, functional groups, representations of Web page, etc.) are defined in a declaratory way, these concepts can be organized into ontologies.

The use of ontologies brings many advantages [7]. First of all generally permitting the multiple heritages, they provide an advantage of expressivity on directed implementations objects. They also allow the creation of communication models of high level, named "peer-to-peer", in which the defined concepts, like domain knowledge, are common to the communicating agents, playing the role of vocabulary shared for the communication between agents.

The use of ontologies in the development of information gathering systems on the Web also increases their flexibility. The entities of a cluster (the domain knowledge) can be defined with the suitable granularity representing the subtle differences in hierarchy between the entities. Lastly, represented in a declarative way, knowledge on the Web pages, and the conditions under which they are considered to represent an example of an entity, is not limited under terms, keywords and statistics. It relates to any fact which can distinguish a class from pages to other classes. For example, such facts can define the structure of page, the areas where to find information suitable to extract and even concepts and the significance of the sentence using of natural language techniques.

3. GENERAL PRESENTATION OF AGATHE ARCHITECTURE

The AGATHE system is a generic software architecture allowing the development of information gathering systems on the Web, for one or a few restricted domains, being this latter advantage a godd enhancement and advantage over MASTER-Web. The AGATHE system is a project developed between France and Brazil [3]. AGATHE implements an information cooperative gathering approach based on software agents that cooperate and exploit ontologies associated to these restricted domains.

In this section, the AGATHE main objectives are first presented, its general architecture and its general functioning. Then the three main subsystems composing AGATHE system are introduced.

3.1 AGATHE objectives

The structure of the Web and its contents evolve continuously. Information appears and disappears, and new concepts and tools are created and change very quickly. The information gathering systems have to be able to follow this evolution. These systems must be extensible, adaptive and flexible devices to exploit information resulting from the Web.

As we already evoked, this software architecture benefits from agents oriented software engineering (which would be extended thereafter to Web services). Such software engineering ensures flexibility and reusability. The starting point of this architecture is a prototype already realized the MASTER-Web system [5]. This system has already adopted the agent approach and also uses ontologies to carry out tasks of classification and extraction of information on the Web on one restricted domain of search.

The AGATHE Architecture reuses the techniques of classification and extraction based on ontologies of MASTER-Web, and deploys them on a complex organization of more effective software agents, with different types of specialized agents in interaction. Moreover AGATHE allows treating several fields of search simultaneously, and has mechanisms of recommendation inter sophisticated fields, and an easier implementation. Lastly, in

its agent oriented implementation, AGATHE respects the recommendations defined by the FIPA [4].

AGATHE is to permit information gathering on restricted fields of the Web that can be gradually widened. For the development of AGATHE, the first restricted domain of search chosen is the academic search domain, more precisely scientific events (international conferences or workshops). The academic research domain concerns relevant information such as call for papers (CFPs), call for participation, title, sponsors place, topics, important dates, program, title of sessions, etc. Information gathering over this domain will then be widened to another restricted domain, the tourism and transport domains, this in order to envisage a displacement related to participation in a particular scientific event (trips, lodging, touristical visits, etc.). With each one of these search domains a specific ontology is associated.

3.2 Architecture and general functioning of AGATHE

The AGATHE general architecture [3], illustrated on figure 1, is articulated around three principal subsystems in interaction:

- (i) *Search subsystem (SSS)* is in charge of querying external search engines on the Web (such as Google) in order to obtain Web pages which will be treated by the ESS,
- (ii) *The Extraction subsystem (ESS)* composed of different "extraction clusters" (EC), each one specialized in the processing of Web pages on a specific field (like that of academic search, or that of tourism),`
- (iii) *The Front Office subsystem (FOSS)* ensures the storage of information extracted from the Web pages treated by the Extraction subsystem, and provides a query interface for the users, which can be humans or other software agents.

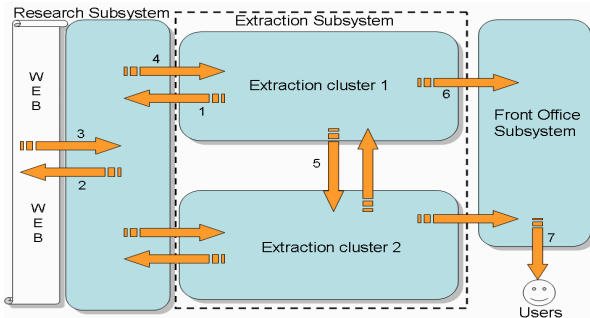


Figure 1. General AGATHE architecture

Each of these three main subsystems of the AGATHE system are multi-agent systems (MAS) composed of software agents with variable degrees of intelligence. Some of them use ontologies to carry out the tasks for which they are conceived. The general functioning of AGATHE is illustrated on figure 1 by the numbered arrows of interaction between its various subsystems:

- 1: A cluster of extraction of the Extraction SubSystem (ESS) requires a search for pages particular to the Search Subsystem;
- 2 and 3: The SSS works like a meta-robot of search, seeking Web pages, by querying existing search engines like Google, Altavista or other;
- 4: These pages are then transmitted to the ESS, more precisely to the agent of the extraction cluster which has done the initial

query (1);

- 5: If necessary, recommendations are sent by the cluster considered to other extraction clusters, this in order to propose pages to them which can potentially interest them;
- 6: Extracted information is transmitted to the Front-Office Subsystem (FOSS), in order to be stored in a specific database, which is accessible by users' queries (7).

3.3 The Search Subsystem (SSS)

Search Subsystem is a MAS composed of three types of agents:

- (i) Search Agents, which search all the Web by the use of traditional search engines (Google, Altavista, Yahoo),
- (ii) Resource Agents which look in specific resource sites of the Web (DBLP, CITESEER,...), and
- (iii) a Supervisor Agent, which creates, deletes, exploits the two previous types of agents, and manages their load.

3.4 The Extraction Subsystem (ESS)

The Extraction Subsystem (ESS) is central in the AGATHE system architecture. Each cluster is associated to one domain ontology. The ESS is a MAS composed of different agents performing different tasks of classification and information extraction, thanks to a domain ontology, and also to an ontology dedicated to AGATHE and named "Agathe ontology".

3.5 The Front Office Subsystem (FOSS)

Front Office Subsystem is the subsystem supporting user interactions with the AGATHE system. The FOSS is composed of two main components: the Mediator and the User Interface. The former has to update the database, from the extracted information transmitted by the Extraction Subsystem (Storage Agent). The latter supports user interactions with the AGATHE system.

This paper focuses on the Extraction subsystem. The following section presents this subsystem in more detail.

4. THE AGATHE EXTRACTION SUBSYSTEM (ESS)

The general aim of this subsystem is to classify Web pages transmitted by the SSS, to extracts relevant information from them and finally to store these pieces of information in a database, to be exploited by the users in the Front Office subsystem. The AGATHE ESS is strongly based on the MASTER-Web system [5]. In the system an agent processes Web pages in performing various tasks of classification and extraction, thanks to domain ontology. The AGATHE system is more ambitious.

In order to permit an information gathering concerning more than one restricted domain, the AGATHE Extraction Subsystem is composed of a set of extraction "clusters" (while MASTER-Web has only one "cluster"). Each of these extraction clusters is related to a specific domain, to which is associated a specific ontology. For example, considering scientific events such as conferences, Call for Papers (CFP) Web pages, can be processed by a cluster related to scientific events, but usually this Web page brings information about trips, hotels, social and cultural events which are simultaneous or with dates near the conference, and so on. Other extraction clusters related to the tourism domain could also process these CFP Web pages.

In order to be more efficient, an extraction cluster is performed by several cooperating software agents, each agent being specialized in a specific task. This distribution in AGATHE allows better performances to treat a very large number of pages. For example, several instances of a same type of agents could share the treatment of these pages running on the same machine or on different machines. This distribution allows distributing the Web page processing on several instances of different extractor agents specialized in the treatment of different parts of the domain ontology related to the cluster. This division is essentially designed for scalability purposes, while in MASTER-Web one agent is responsible for a class of pages (like CFPs, for instance) and could not scale to a better performance.

The following subsections describe in detail each type of agent composing this ESS, in particular the tasks that they perform.

4.1 Agents and ontologies used

Extraction Subsystem has to (i) generate queries of Web pages to the Search Subsystem, (ii), and to treat results of these queries (the Web pages that the Search Subsystem has found out). This latter task is the main task, and consists in the subtasks of page validation, functional classification, and information extraction.

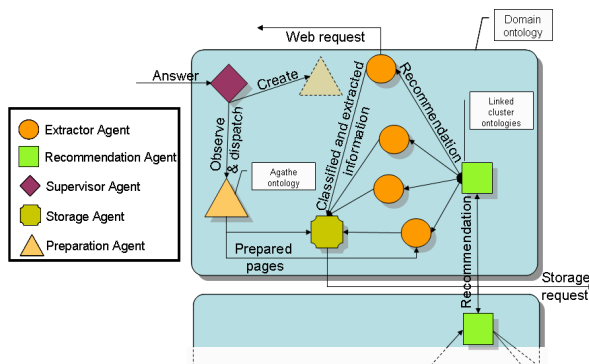


Figure 2. Internal architecture of an Extraction Cluster.

As illustrated in figure 2, each of these Extraction Clusters is a multi-agent system performing the classification of the Web pages, and information extraction on these pages. The various agents that compose the cluster are: a set of Extractor Agents and Preparation Agents, a Supervisor Agent, a Recommendation Agent, and a Storage Agent. These agents perform specific tasks in the extraction cluster. Some of these agents use ontologies to perform their tasks. Both defined ontologies were developed using the Frames formalism under the Protégé environment [15].

Domain ontologies describe the restricted domain considered for information gathering. Each extraction cluster is associated to such ontology. Figure 3 presents a part of domain ontology related to the academic research field. This part concerns the live scientific events, concerning publication, CFP (Call For Papers).

The internal ontology, the Agathe ontology, specifies mainly some concepts used by AGATHE and MASTER-Web for the classification and extraction tasks performed on Web pages. Figure 4 presents a subset of this ontology with the concept of Web page and two specific concepts for information extraction (Slot-Recognizer and Slot-Extractor).

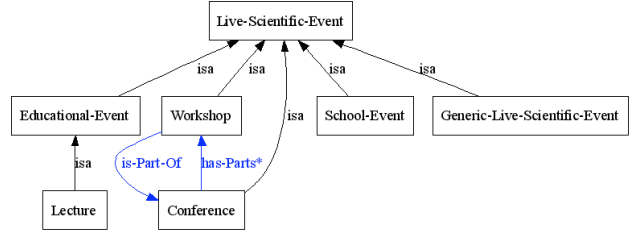


Figure 3. A part of a domain ontology about Science describing "scientific events"

All agents of the extraction subsystem are cognitive agents and employ the Jess inference engine in their reasoning for the tasks of classification and extraction. The behaviors of these agents are mainly specified by production rules. Most of them exploit ontologies using the Protégé plug-in JessTab. The following subsections present each of these agents in detail.

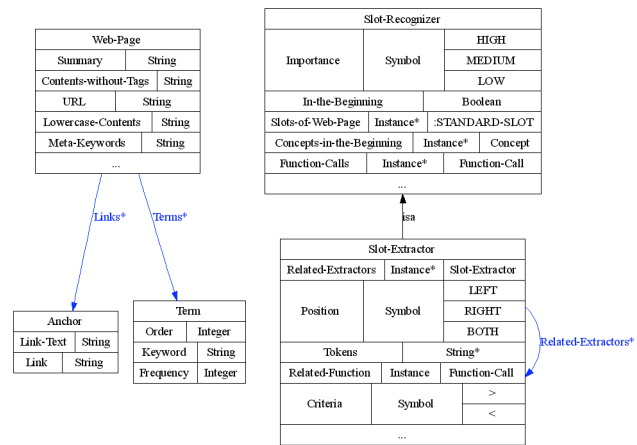


Figure 4. Classes, slots and relations of the Agathe ontology

4.2 Preparation Agents

The Preparation Agents receive Web pages from the Search Subsystem and performs some treatment on them, that will be described below. These agents are created by the Supervisor Agent of the considered extraction cluster and are deleted by this same agent when they are not being used any more. These agents perform the first treatments, thus permitting more easily to exploit Web pages. These treatments are based in the treatment performed by MASTER-Web agents, which are explained below (figure 5):

- *Validation.* This task verifies if Web pages obtained are in HTML format, accessible, and if they are already stored in the database. Pages that do not meet these requirements will not be considered in following treatments.
- *Pre-processing.* This task identifies the content, the title, the links, and the email from the Web pages, using techniques of information retrieval and eventually, if necessary, natural language techniques.
- *Functional classification.* This task is knowledge-based and uses the Jess inference engine exploiting the Agathe ontology. Thanks to a specific knowledge base (production rules), the Preparation Agent uses this ontology to classify Web pages according to a functional aspect. The functional categories in which the pages will be classified are: messages, lists of links to potentially useful pages (e.g. a list of CFPs), auxiliary pages

(pages that contain some pieces of information but don't represent an instance of an entity, e.g. a separate page of topics of a conference which has its own page), pages selected for extraction, and finally pages considered as invalid.

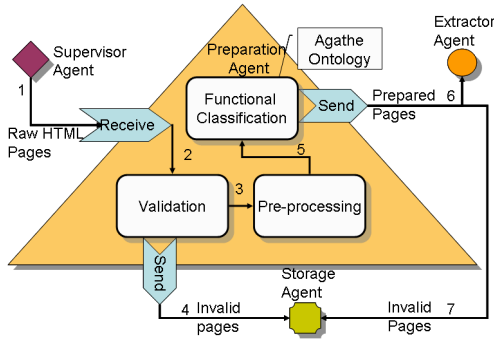


Figure 5. A Preparation Agent and its internal tasks

Indeed, to achieve more performance and flexibility, some of these treatments considered as generic, have been extracted from the original MASTER-Web agents. Then, it is now possible to duplicate and migrate such agents when the load is too high.

4.3 Extractor Agents

The Extraction Cluster is composed of several Extractor Agents (figure 6) associated to a specific domain ontology linked to the cluster. The task of these agents is to perform a *semantic classification* and then *information extraction* over the Web pages that they receive from the Preparation Agents.

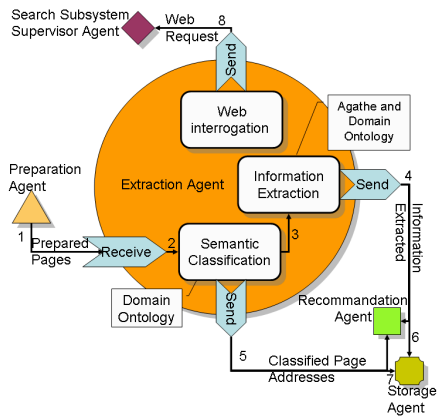


Figure 6. An Extraction agent and its internal tasks.

Each Extractor Agent is associated to a particular class (concept) of domain ontology. Like the Preparation Agent, the Extractor Agent is based on the extraction and content classification of the MASTER-Web agent, and uses Jess and JessTab to benefit from ontologies. For instance, within the domain of academic science, sub-domain of scientific events, a particular Extractor Agent is associated to the concept or class "Call For Papers" of the ontology, and performs filtering and information extraction in the Web pages it receives.

Depending on the classification results for a Web page, and referring to ontologies, the Extractor agent performs information extraction. Extractor Agents may also classify pages. For example, agent defined as "Call For Papers" agent could classify different calls for papers for conferences, journals, book chapters

and many other classes from Science ontology. Some specific information, related to these classes is extracted. Extracted information is then transmitted to the Storage Agent.

4.4 Recommendation Agent

The Recommendation Agent (figure 7) receives prepared pages from the Preparation Agent and dispatches them to other agents in the same cluster or to other clusters. It performs three main tasks:

- (i) *Internal recommendation*: it recommends pages/links that have some interest to other Extractor Agents of the cluster.
- (ii) *External recommendation*: it recommends some pages/links to other Extraction Clusters, pages that could be interesting for them (an example is given just after the next figure). For this task, the Recommendation Agent has to know ontologies of the different clusters involved. It dispatches these pages to the various Recommendation Agents of the cluster concerned.
- (iii) *Dispatching*: it dispatches pages that have been recommended by another Recommendation Agent of other Cluster.

Two clusters are linked if a contextual link exists between them. For instance, the agent responsible for scientific events can suggest for a Tourism cluster some information found on "call for papers" pages, which is related to accommodation and transport facilities to participate in such scientific events.

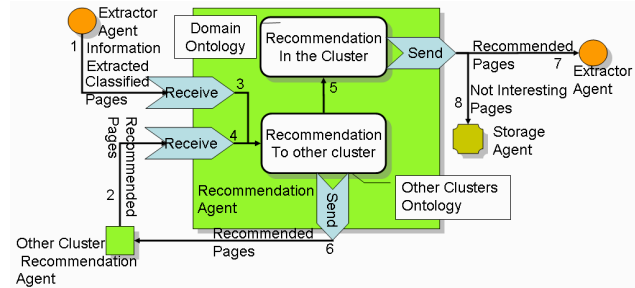


Figure 7. Recommendation agent and its internal tasks.

4.5 Storage Agent

The Storage Agent is in charge of storing the extracted/classified information in the database of the Front Office Subsystem. This agent prepares and performs the storage. It treats this information so as to conform to the storage format, according to the storage structure of the database. Then it stores the classification results and the extracted information in the database to be exploited in the Front Office Subsystem by the users.

4.6 Supervisor Agent

The Supervisor Agent of the Extraction Cluster has functionalities similar to one of the Supervisor Agent of the Search Subsystem. It supervises the Preparation Agents activities. First of all, it receives the queries' results from the Search Subsystem, then it creates one or more Preparation Agents that will treat these results before transmitting them to the Recommendation Agent of the cluster considered.

5. IMPLEMENTATION DETAILS AND FIRST RESULTS

The AGATHE system is currently under development and this section presents details of its implementation and first results.

5.1 Implementation details

The AGATHE architecture is deployed in the Eclipse environment in Java, and uses the Jade multi-agent platform [9]. Currently the Search Subsystem (SSS) and the Front Office subsystem (FOSS) are composed of agents developed in Java. In the Extraction subsystem (ESS), the agents that use a domain ontology and/or the Agathe ontology to perform specific tasks, are developed with the Jess inference engine [10]. Currently, the Extraction subsystem works over only one extraction cluster without recommendation mechanism.

For the construction and the handling of ontologies, the Protégé environment [15] is used, and the exploitation of the ontologies by the Jess agents is done via the component JessTab [2]. The classification results and information extracted are stored in a MySQL relational database system.

5.2 First results

To develop and test the AGATHE architecture we use the restricted domain of the scientific events in academic research. For this, a domain ontology has been reused from MASTER-Web. The prototype first performs a gathering of pages concerning Calls For Paper (CFP), a filtering of the obtained pages, then it classifies them into 8 CFP subclasses (conference, workshop, journal etc) and finally it extracts relevant information and stores it into a database.

On a sample of 310 Web pages obtained from the Web by a search engine, the AGATHE prototype correctly classifies 280 pages (90,32%). On these 280 pages, only 81 relevant pages are treated in information extraction.

6. CONCLUSION

While being limited to restricted domains, our research hypothesis is that taking into account information retrieval contexts is possible and must lead to more relevant information gathering. In this paper, first, a restricted domain and cooperative information gathering approach, based on software agents and ontologies has been proposed. Then, a generic software architecture, AGATHE, permitting the development of such gathering systems has been presented in detail, and makes several enhancements over an early implementation, the MASTER-Web system.

The AGATHE architecture is composed of three main subsystems in interaction, a Search subsystem, an Extraction subsystem that ensures a classification and an extraction of information of the Web pages, and finally a Front Office subsystem that stores this extracted information in a relational database and allows users to exploit it. Each of these subsystems is composed of software agents exploiting, for some, a specific ontology, called AGATHE ontology, or an ontology related to the restricted domain(s) considered, to perform the tasks of classification, filtering, recommendation or extraction of information using reasoning.

The AGATHE system is currently under development, with a first prototype running and first results beginning to be produced. For the improvement of this prototype, three axes of evolution are currently considered:

- (i) the integration of techniques of natural languages processing in particular for the tasks of classification and information extraction, in order to make them more powerful,

- (ii) the integration of learning techniques in particular for the recommendation tasks, and finally
- (iii) the use of Web services (WS), perceived as components which can be used to develop some informational agents. On this last point, the WS library defined for the travel industry in the Satine project [16], could be used in a forthcoming version of AGATHE.

7. REFERENCES

- [1] Ambite J., Knoblock C., « Agents for information gathering ». In Software Agents. Bradshaw, J. (ed.), MIT Press, Pittsburgh, PA, USA, 1997.
- [2] Eriksson H., « Using JessTab to Integrate Protégé and Jess », IEEE Intelligent Systems, 18(2):43-50, 2003.
- [3] Espinasse B., Fournier S., Freitas F., « AGATHE: an Agent and Ontology based System for Restricted-Domain Information Gathering on the Web », Proceedings of the International Conference on Research Challenges in Information Sciences (IEEE-RCIS), April 23-26, Ouarzazate, Morocco, 2007.
- [4] FIPA, IEEE Foundation for Intelligent Physical Agents, <http://www.fipa.org/>, 2000.
- [5] Freitas, F. Bittencourt, G. « An Ontology-Based Architecture for Cooperative Information Agents », Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 2003.
- [6] Gaizauskas R., Robertson A., « Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web ». Proceedings of RIAO, Computer-Assisted Information Search on the Internet. Montreal, Canada. 1997.
- [7] Gruber, T. R., « Towards Principles for the Design of Ontologies Used for Knowledge Sharing », Int. Journal of Human and Computer Studies, 43(5/6): 907-928. 1995.
- [8] Huhns, M. N., Singh, M. P., « Distributed Artificial Intelligence for Information Systems », CKBS-94 Tutorial, June 15, University of Keele, UK. 1994.
- [9] Jade, Jade tutorial <http://jade.tilab.com>, 2006.
- [10] Jess, <http://hezberg.casandia.gov/Jess>, 2006.
- [11] Kushmerick N., « Gleaning the Web ». IEEE Intelligent Systems. EUA 1999.
- [12] Muslea I., « Extraction Patterns for Information Extraction Tasks: A Survey ». www.isi.edu/~muslea/RISE/ML4IE/ml4ie.muslea.ps, 1999.
- [13] Nwana N.H., « Software Agents: An Overview », Knowledge Engineering Review, Vol. 11, No 3, pp.1-40, Sept 1996. Cambridge University Press, 1996.
- [14] Oates T. et al, « Cooperative Information Gathering: A Distributed Problem Solving Approach ». Computer Science Technical Report, University of Massachusetts, 1994.
- [15] Protégé, <http://Protégé.stanford.edu/>, 2006
- [16] Satine project, <http://www.srdc.metu.edu.tr/webpage/projects/satine/>, 2001.