AGATHE: 
An Agent- and Ontology-Based System for Gathering Information about Restricted Web Domains

Bernard Espinasse, Domaine Universitaire de St Jerome, France
Sébastien Fournier, Domaine Universitaire de St Jerome, France
Fred Freitas, Universidade Federal de, Brazil

ABSTRACT

Due to Web size and diversity of information, relevant information gathering on the Web turns out to be a highly complex task. The main problem with most information retrieval approaches is neglecting pages’ context, given their inner deficiency: search engines are based on keyword indexing which cannot capture context. Considering restricted domains, taking into account contexts may lead to more relevant and accurate information gathering. In the last years, we have conducted research with this hypothesis, and proposed an agent- and ontology-based restricted-domain cooperative information gathering approach accordingly; that permit the development of specific information gathering systems. In this article, we present this novel approach based on these guiding ideas, and a generic software architecture, named AGATHE, which is a full-fledged scalable multi-agent system. [Article copies are available for purchase from InfoSci-on-Demand.com]

Keywords: Agent Software; Information Retrieval; Ontologies

INTRODUCTION

Because of the size of the Web and the diversity of accessible information, to gather relevant information from the Web turns out to be a highly complex task. Without taking explicitly into account the search context, the majority of the current approaches of information retrieval (IR) let escape many forms of organized information of the Web, for example, specific domains or “clusters” of information.

However, the field known as Symbolic Artificial Intelligence (AI) has faced a similar challenge in the past. During the Seventies, researchers from this field tried to produce systems that could cope with inference capabilities about everything. The lesson learned (Newell et al., 1959) was that the use of knowledge-based systems is feasible only over restricted domains,
which led to the relative success of the expert systems. This policy is also valid for the IR field. Indeed, the evaluation of the IR systems is mainly carried out over homogeneous corpora, whose texts relates to only one subject and often come from the same source, and not from text sets with diverse contents and writing styles, as it is the case of those available on the Web. This fact is also besides at the origin of the development in IR of specialized search engines (Steel, 2001; McCallum et al., 1999).

Another argument pleading for a restricted domain in IR relates to Information Extraction (IE). Generally, IE works over textual documents collections (Muslea, 1999; Embley et al., 1998). The task consists in extracting data starting from specific classes of Web pages (Gaizauskas & Robertson, 1997). It concerns the identification of specific fragments from a document, which should constitute the core of its semantic contents (Kushmeric, 1999). The main goal of IE is to populate databases about specific domains - such as Tourism, Academia, etc -regrouping information coming from many Web pages spread over geographically distributed sites. These databases save users’ work on finding, checking and comparing the data which then can be easily queried by users.

Taking such a specific domain context into account enables better data processing (Etzioni et al., 2004). It is the case of the extraction of majority of information from a given class of pages (for example the value of the dollar from a currency exchange rates page, subjects of interest of a researcher from his homepage and so on). Another advantage is to make possible for the users to carry out queries combining in particular search keys relative to various classes of pages, allowing complex requests (the search of the papers published in a certain whole of conferences for example). Thus, it is possible to build sophisticated applications in order to gather Web information from specific domains. With the “Tourism” cluster for example, applications could retrieve, extract and classify data about hotels, passage tickets, and cultural events.

In the last years, we have conducted research with these research hypotheses, and produced ontology-based restricted-domain cooperative information gathering multi-agents accordingly, that permit the development of a specific information gathering systems e.g. the MASTER-Web system (Freitas & Bittencourt, 2003). In this article, we present a novel approach based on these guiding ideas, a generic software architecture, named AGATHE, which is a full-fledged scalable multi-agent system, which includes a better-designed and organized agent topology. The article is organized as follows: in section 2, we introduce the concept of cooperative information gathering, and the interest of using agents and different kinds of ontologies to develop intelligent gathering systems on one or more restricted domains of the Web. Section 3 introduces the AGATHE system, a multi-agent architecture for development of such intelligent gathering systems on the Web, its objectives, its architecture with its three main subsystems, and its general functioning. AGATHE is inspired from an early prototype developed in Brazil and named MASTER-Web (Freitas & Bittencourt, 2003). Sections 4 to 6 present in detail these subsystems composing AGATHE system: the Search Subsystem, the Extraction Subsystem and the User Subsystem. Section 7 presents some implementation details of the prototype in progress and some results. Finally, we conclude with some research perspectives.

COOPERATIVE INFORMATION GATHERING AND ONTOLOGIES

Suggested by Oates et al. (1994), the concept of “Cooperative Information Gathering” (CIG), is based on the distributed problem solving paradigm for the fields of multi-agent systems (MAS) and distributed artificial intelligence (DAI) (Huhns & Singh, 1994) (Nwana, 1996). 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 to discover relevant clusters of information.

Other research works recommend the use of agents for information gathering. In Ambite and Knoblock (1997), in hierarchical classes the databases of a large numerical library, each class has its own agent with explicit knowledge about it. These agents build the research plans, which improve the effectiveness in the search process. Using such a tool on the Web requires a correct pairing of the pages discovered on the Web with these classes and to extract information from it to feed these databases.

Decker, Nagendra Prasad and Wagner (1995) have proposed MACRON, an agent architecture adapted to Cooperative Information Gathering. In MACRON the top level user queries drive the creation of partially elaborated information gathering plans, resulting in the employment of multiple cooperative agents for the purpose of achieving goals and subgoals within those plans. MACRON is composed of three types of autonomous agents: reasoning agents, low level retrieval agents, and user interface agents.

More recently, Lesser, Horling, Klassner, Raja, Wagner and XQ Zhang S., (2000) have proposed the BIG system. BIG is an informational agent, that plans to gather information to support a decision process, reasons about the resource trade-offs of different possible gathering approaches, extracts information from both unstructured and structured documents, and uses the extracted information to refine its search and processing activities.

Interest of Ontologies for CIG

In order to take into account context for domain restricted CIGs, it is necessary to use knowledge related to the concerned domain. The use of ontologies in CIGs is justified by the advantages of using declarative solutions, due to a number of reasons.

First of all, declarative solutions provide closer integration of an ontological approach with a more direct translation of the domain knowledge.

Moreover, the tasks of extraction and classification on the Web, which deals with unstructured or semi-structured data, require frequent changes of theirs solutions. With declarative knowledge, defined in ontologies, such changes can be easily taken into account, without the needs of recompiling code or stop execution. In this way, the use of ontologies constitutes a notable advantage of extensibility. In more of the possibilities of inferences, concepts implied in these tasks (for example cluster entities, functional groups, representations of Web page, etc.) are defined in a declarative way in ontologies.

The use of ontologies brings many others advantages (Gruber, 1995). They permit multiple inheritances and take advantage of expressivity in comparison to using object-oriented implementations. They also enable the use of high level communication models, in which the defined concepts, like domain knowledge, are common to the communicating agents, playing the role of shared vocabulary for agents’ communication. Finally, the use of ontology increases the flexibility of information gathering systems.

Different Kinds of Ontologies for CIG

In information gathering over restricted domains of the Web, the major tasks to perform are Web pages’ retrieval, classification and information extraction. For the realization of these tasks, three types of ontologies can be used in a complementary way:

Domain ontologies: these are one or more ontologies related to the restricted domain. These ontologies should cover the concepts, relations, restrictions, terminology and valid axioms of the domain and can be used, e.g., to classify Web pages and extract relevant information from them.

Linguistic ontologies: the integration of natural languages processing techniques exploiting linguistic ontologies is relevant in
particular for the tasks of classification and information extraction, in order to make them more powerful, in particular in clearing up a lot of ambiguity related to natural languages. Typical examples are co-reference resolution (e.g., to know at whom the pronoun “it” refers to in the phrases “My dog likes my cat. It purrs every morning to wake me up.”) and passive voice phrases where simple extractors, such as wrappers (Kushmerrick, 1999) fail. Wordnet (Miller, 1995) and the ontologies from the GATE project (General Architecture for Text Extraction) are good examples of ontologies of this type.

Operational ontologies: they gather and organize knowledge used by a software tool, enabling this tool to perform the set of tasks for which it was designed. In such a tool, the main interest of using knowledge defined in an operational ontology is still related to its declarativity. This declarativity brings a greater extensibility to the tool by allowing many possibilities of evolution in the realization of its tasks.

Classification and information extraction tasks concerning semi- or unstructured information are very difficult to realize. They use, in general, heuristics, often developed in an empirical way, and require, in consequence, many adjustments. The exploitation of declarative knowledge defined in an operational ontology promises to facilitate the realization of these tasks. Such operational ontologies concern knowledge that is not specific to the restricted domain considered, but associated to the manner to exploit Web page related to this domain, in particular in classification and extraction tasks. Such instrumental knowledge is not limited to terms, keywords and statistics like is usually done in common gathering systems. This knowledge can concern any fact that make possible, in page classification, to distinguish a class of pages from other classes, or in information extraction, to consider the structure of the page treated, of the probable areas of this page where to find suitable information to extract.

AGATHE OVERVIEW

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 good enhancement and advantage over MASTER-Web (Freitas & Bittencourt, 2003). The AGATHE system is a project developed between France and Brazil (Espinasse et al., 2007), in which cooperative agents exploit one (or more) domain ontologies related to one (or more) restricted domains and an operational ontology to perform its various processing tasks over Web pages in a distributed and cooperative way.

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, and some implementation details are finally given.

AGATHE Objectives

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 (Freitas & Bittencourt, 2003). This system has only one agent which 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 from MASTER-Web, and deploys them onto a complex and distributed organization of more effective software agents, with different types of specialized agents in interaction. Moreover, AGATHE allows for treating several fields of search simultaneously and has mechanisms of recommendation inter sophisticated fields, with an easier implementation. Lastly, in its agent oriented implementa-
tion, AGATHE respects the recommendations defined by the FIPA (FIPA, 2000).

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 about events, such as title, sponsors, place, topics, important dates, program, title of sessions, etc. contained in calls for papers (CFPs) and calls for participation pages. Information gathering over this domain will then be widened to another restricted domain. For instance, we can consider the tourism and transport domains in order to envisage a displacement related to participation in a particular scientific event (trips, lodging, touristic visits, etc.). With each of these search domains there is a specific ontology associated.

Architecture and General Functioning of AGATHE

The AGATHE general architecture (Espinasse et al., 2007), illustrated on figure 1, is articulated around three principal subsystems in interaction: the Search subsystem (SSS), the Extraction subsystem (ESS), and the User subsystem (USS).

Each of these three main AGATHE subsystems of the system are multi-agent systems (MAS) themselves, composed of software agents. Some of them use ontologies to carry out the tasks for which they were conceived.

The Search subsystem (SSS) is in charge of querying external Web search engines (such as Google) in order to obtain Web pages which will be treated by the Extraction subsystem (ESS). This Subsystem is a multi-agent system (MAS), composed of different types of agents to search all the Web by the use of traditional search engines (Google, Altavista, Yahoo), to look in specific resource sites of the Web (DBLP, CITESEER, ...), ...

The Extraction subsystem (ESS) is the heart of the whole architecture and is 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). Each cluster is associated to one domain ontology. This subsystem is a MAS composed of different agents performing different tasks of classification and information extraction, supported by different ontologies.

The User subsystem (USS) performs the storage of information extracted from the Web pages already treated by the Extraction subsystem, and provides a query interface for the users, which can be humans or other software agents. This subsystem is in charge of the user interactions within the AGATHE system.

The general functioning of AGATHE is illustrated on figure 1. Below is description of the numbered arrows that represents the step-by-step interactions among its various constituent 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 to other extraction clusters, in order to propose pages to them which can potentially interest them;
- 6: Extracted information is then transmitted to the Front-Office Subsystem (FOSS), in order to be stored in a specific database, which is accessible by users’ queries (7).

Implementation Details

The AGATHE system is currently under development between France and Brazil. AGATHE
Figure 1. General AGATHE architecture

is deployed in the Eclipse environment in Java, and uses the Jade multi-agent platform (Jade, 2006). The Search Subsystem (SSS) and the User Subsystem (USS) are composed of agents developed in Java. In the Extraction subsystem (ESS), the agents using the domain ontology and/or the operational ontology to perform specific tasks, are developed with the Jess inference engine (Jess, 2006). Currently, the Extraction subsystem works over only one extraction cluster without recommendation mechanism.

For the construction and the handling of ontologies, defined by Frames, the Protégé environment (Protégé, 2006) is used, and the exploitation of the ontologies by the Jess agents is done via the Protégé plugin JessTab (Eriksso, 2003).

The classification results and information extracted are stored in a MySQL relational management database system according the RDF Format. To store in this data base these classification results and extracted information in RDF format, task performed by the Storage Agent of the Extraction Subsystem, and to exploit this database in the User Subsystem, the Jena framework (Jena, 2006) is used. Jena is a Java framework for building Semantic Web applications providing a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine.

The following section presents in detail the three subsystems composing AGATHE.

THE AGATHE SEARCH SUBSYSTEM

As illustrated in figure 2, the Search Subsystem receives requests for Web pages from specific agents of a cluster. For instance, the Articles agent from the Science cluster asks for pages that contain keywords such as “introduction”, “related work”, “conclusion”, and some others. Then the Search Subsystems forwards this query to existing search engines, gathers the Web pages returned by them and return these pages to the specific Extractor Agents from the Extractor Cluster which solicited them.
Three types of agents contribute to the information search process: (i) Search Agents, (ii) Resource Agents and (iii) Supervisor Agent. The Search Agent performs requests to existing search engines (Google, Altavista, Yahoo, …). It receives requests from an specific Extractor Agent. After having merged the different results obtained by the various engines, the Search Agent directly transmits them to the Extraction Subsystem.

The Resource Agent is similar to a Search Agent, but it performs requests only to specialized resources of the Web. For example, for information related to the academic research, such resources can be the CITESEER site for publications, the DBLP site for authors, a specific web service, or a specialized database. It is requested by the Supervisor Agent, and it transmits its results directly to a specific Extractor agent in the Extraction Subsystem.

The Supervisor Agent coordinates the activities of the Search Subsystem. It receives requests from the Extraction Subsystem and, aware of the workload of various Search and Resource Agents, it manages for the best allocation of these requests to be treated. According to the strain over these agents, it creates (or even deletes) Search and Resource Agents. For performance reasons, the Supervisor Agent can also decide to move these Search Agents to CPUs with less work. The Supervisor Agent manages also subscriptions, permitting the Extraction Subsystem to receive periodically results from a specific information request. The strategy used can be called “push” and the results are directly transmitted, without the necessity of formulating further requests.

THE AGATHE EXTRACTION SUBSYSTEM (ESS)

The general aim of the Extraction Subsystem is to classify Web pages transmitted by the Search subsystem, and to extract relevant information from them and finally to store this information in a database, to be accessed by the users in the User Subsystem.

In the following sections, we present the architecture of the Extraction Subsystem, the different ontologies used in this Subsystem to perform classification and extraction tasks, the way that agents employ these ontologies, and finally, we present in details each agent of this subsystem.
Extraction Subsystem Architecture

The AGATHE ESS is strongly based on the MASTER-Web system (Freitas & Bittencourt, 2003). In the MASTER-Web system each agent processes Web pages performing the various tasks of classification and extraction using ontologies. On its turn, 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 processes only one cluster at a time). Each of these extraction clusters is related to a specific domain, to which is associated a specific ontology. For example, considering scientific events deployed via Call for Papers (CFP) Web pages, these can be processed by classes of an academic research cluster, classes which are related to scientific events. However these Web pages usually bring 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, like, searching the web for useful pages, pre-processing, extracting, supervising, recommending and storing. This distribution in AGATHE allows for a better performance when treating a very large number of pages. For example, several instances of a same type of agent 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 when a high number of pages have to be processed.

The Extraction Subsystem’s functions are twofold:

i. to ask for Web pages to the Search Subsystem, and

ii. to process these pages (the Web pages that the Search Subsystem has deployed, which are supposed to belong to the class being processed by the Extractor agent which asked for them).

This latter task constitutes the backbone of the whole system, and consists in the subtasks of page validation, functional classification, and information extraction.

As illustrated in figure 3, each of these Extraction Clusters is a multi-agent system performing the classification of the Web pages, and information extraction from these pages. The various agents that compose the cluster are:

- a set of Extractor Agents and
- a set of Preparation Agents,
- one Supervisor Agent,
- one Recommendation Agent, and
- one or more Storage Agents.

These agents perform specific tasks in the extraction cluster. Some of these agents use ontologies to perform their tasks.

Figure 4 illustrates with a UML sequence diagram the basic interactions between these agents of a same extraction cluster. At the first phase, the Extractor agent sends a query (for example, “call for paper”) to receive pages from the search engines, and the query result is sent to the Supervisor agent. This agent creates one or more Preparation agents to represent the pages so that the Extraction agent could reason with it and decides to which class it belongs to and extract data from it. During this preparation, a validation process is carried out, which basically checks whether a page is in a format processable by the system. Valid pages are then transmitted to the Extractor agent while the invalid ones are stored in database via a Storage agent. Classified pages and extracted data are sent to this agent too and to a Recommendation agent, which may
Figure 3. Internal architecture of an extraction cluster

Figure 4. Interactions between extraction cluster’s agents

send pages or links to other Extractor agents if they are interested in.

Before presenting in detail these different agents composing the ESS, the following subsections introduce the different ontologies used by these agents and how they benefit from them.

Ontologies Used in the Extraction Subsystem

The cooperative agents composing the AG-ATHE Extraction Subsystem exploit two types of ontology to perform tasks of pages classification and information extraction. The first one is
an internal ontology, an operational ontology, called “Agathe ontology”, used to perform in a cooperative way, various information treatments on the Web pages. The second type are the domain ontology(ies), related to one or more restricted domain of search.

The current version of AGATHE does not use Linguistic ontologies. The information extraction process is based only on the presence or absence of concepts belonging to the domain ontology. An upcoming version of AGATHE will use linguistic ontologies and natural languages processing techniques to improve classification and extraction tasks.

A domain ontology concerns the restricted domain considered for information gathering. Each extraction cluster is associated to such ontology. Figure 5 presents a part of the domain ontology, named “Science ontology”, relative to the academic research field. This part concerns the live scientific events, concerning publication, CFP (Call For Papers).

The operational Agate ontology is already defined and used in MASTER-Web system and named inside “Web ontology”. This ontology specifies the main concepts used by AGATHE for the classification and extraction tasks performed on Web pages. Figure 6 presents a subset of this ontology, related to the concept of Web page and two specific concepts for information extraction (Slot-Recognizer and Slot-Extractor).

Both kinds of ontologies are defined in Frames using the Protégé environment (Protégé, 2006).

How Agents Use Ontologies

Agents of the Extraction Subsystem are cognitive agents and employ the Jess inference engine (Jess, 2006) in their reasoning for the tasks of classification and extraction, using production rules written in Jess. The ontologies specified in the latter section are translated to Jess facts thanks to the JessTab (Eriksson, 2003) Protégé plugin. The general structure of such rules is:

1. Name of the rule
2. Precondition: presence in the facts base of concepts that belong to the operational Agathe ontology and associated attributes
3. Test on attributes obtained (begin by keyword test).
4. Action of the rule specific to the rule (begin by => symbol).

Here is an example of a specific extraction rule, in Jess/JessTab syntax, used by the Extractor agents:

(defrule r_454
  (object (Page-Status STORED) (is-a Processing-Monitor))
  (?f<- (object (Importance MEDIUM) (is-a Slot-Recognizer))

Figure 5. A part of the “Science ontology” concerning scientific events

![Diagram](image-url)
Figure 6. Main classes, slots and relations of the operational ontology

(Slot-in-Process ?s)
(Concepts $?cb&:(> (length$ $?cb) 0))
(Absent-Concepts $?ac)
(not (object (is-a Slot-Found) (Slot-in-Process ?s)))
(test and
 (= (count-occurs-once (words-of-concepts $?ac) (slot-get [LINKS-TEXTS] Values)) 0)
 (> (count-occurs-once (words-of-concepts $?cb) (slot-get [LINKS-TEXTS] Values)) 0)))
=>
(slot-insert$ [SLOTS-FOUND] Instance-Values ?s))

It tests if the slot ?s has already been found (the condition (not (object (is-a Slot-Found) (Slot-in-Process ?s)) ), and then, in the test part of the rule, it checks whether one of the concepts that should be present (S?cb, in the rule) in the text of a Web link and whether the absent concepts are not present (S?ca). If these conditions are met, the slot ?s is found so its name is stored in a list of slots found (the fact [SLOT-FOUND]). Note that rules used by agent referencing ontologies can be complex to develop, consequently a specific tool has been developed to permit to create them more easily.

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

Preparation Agents

Preparation Agents receive Web pages from the Search Subsystem and perform 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 reason with the Web pages. These preparation tasks are based in the treatment performed by any MASTER-Web agent, which are explained (figure 7):

Validation. This task verifies if the 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 collects contents (in various representations, for example, without stopwords, lowercase, without tags, etc), title, links, and emails from the Web pages, using information retrieval techniques and, 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 7. A Preparation Agent and its internal tasks in order to achieve better performance and flexibility, it is possible to duplicate such preparation agents to reduce their strain.

The Extractor Agents

The Extraction Cluster is composed of several Extractor Agents (figure 8) associated to a specific domain ontology linked to the cluster. The task of these agents is to perform a semantic classification over the Web pages that they

![Figure 7. A preparation agent and its internal tasks](image-url)
receive from the Preparation Agents and then extract information from these pages.

Each Extractor Agent is associated to a particular class (concept) of the domain ontology. The Extractor Agent is based on the extraction and content classification of the MASTER-Web agent, and uses also Jess and JessTab to benefit from ontologies. For instance, within the domain of academic science, sub-domain of, a particular Extractor Agent is associated to the concept or class scientific events of the Agathe ontology. After it recognizes a page as representing an instance from this class, it performs filtering and information extraction.

Depending on the classification results for a Web page, and making use of the ontologies, the Extractor agent performs the information extraction. For example, the “Call For Papers” agent could classify different calls for papers for conferences, journals, book chapters and many other classes defined in the Science ontology (domain ontology), which are subclasses of the class Scientific-Events. After finishing the tasks, the extracted information is then transmitted to the Storage Agent.

**The Recommendation Agent**

The Recommendation Agent (figure 9) receives prepared pages from the Preparation Agent and dispatches them to other agents in the same cluster or to other clusters. It accomplishes three main tasks:

- **Internal recommendation**: it recommends pages/links that have some interest to other Extractor Agents of the cluster.
- **External recommendation**: it recommends some pages/links to other Extraction Clusters, pages that could be interesting for them. For this task, the Recommendation Agent needs
Figure 9. Recommendation agent and its internal tasks

The Storage Agent

The Storage Agent is in charge of storing the extracted/classified information in the database of the User Subsystem. This agent prepares and performs the storage. It treats the received information so as to conform to the storage formats, according to the storage structure of the databases tables. It also saves the classification results and the extracted information in the database to be queried by the users via the User Subsystem.

More precisely this Storage stores in persistent memory the instances of the class “Slot-Found” of the Agathe ontology, class where the extracted information is stored by the Extraction Agent, and store these information in the RDF format in a relational database. For this task, this agent uses the JENA (Jena, 2006) environment facilities. We use RDF format in order to work at knowledge level and to keep semantic information related to the ontology.

THE AGATHE USER SUBSYSTEM

The User Subsystem (USS) is the subsystem supporting user interactions with the AGATHE system. The USS is composed of two main components (Figure 10).

The first component concerns the coherence checking of the extracted information stored in the data base. Developed in the JENA environment and SPARQL language, it permits to the user, according an interactive way, to detect incoherence and update consequently the RDF data base of extracted information. Indeed, the extraction process is not always perfect, and
some incoherencies can appear. Integration of techniques of natural languages processing in the information extraction task could reduce these incoherencies.

The second component supports user interactions with the AGATHE system to exploit with queries the extracted information stored in the data base. The SPARQL language and SQL language can be used for it.

**FIRST RESULTS**

The AGATHE system is currently under development between France and Brazil. To develop and test the AGATHE architecture, the restricted domain of the scientific events in academic research has been chosen. The prototype first performs a gathering of pages concerning Calls For Paper (CFP), then it filters these pages and classifies them into 8 CFP subclasses (conference, workshop, journal etc). Finally, it extracts relevant information and stores them into a database. The classification and extraction results are stored in instances of the Slot-Found class of the operational Agathe ontology (figure 11). Most important attributes (slots) of these instances have as values these results:

- **Slots-of-Web-Page:** this attribute defines the slot of the Web-Page class in which the instance has been found,
- **Values:** defines the content that appears in the Web page,
- **Start-Position:** is the starting character of the value in the page,
- **Slot-in-Process:** define the slot of the class at which the instance fits.

The attributes (slots) of this Slot-Found class are presented in figure 11:

The classification and extraction results are stored in instances of the Slot-Found class of the Agathe ontology. For example, the Web page presented in figure 12 has been classified in the “Conference” class by AGATHE system using the domain ontology and the Agathe ontology.

Figure 13 presents AGATHE classification and extraction results concerning the Web page shown at figure 12. These results are stored in an RDF database by the Storage agent.

The AGATHE system, using the domain and Web ontologies, has first classified this Web page in the “Conference” class, and then extracts various pieces of information, such as “30 April 2007”, instance of the “Initial-Date” class. This information starts at the character 4722 (start-Position) of the slot “Lowercase-Contents” of the Web-Page class. The “Lowercase-Contents” being a working formulation of the Web page treated stored in the database. This information could permit a new extraction process, for example to extract information concerning “Registration-Fees” related to this conference. This new extraction process could be based on natural languages processing techniques.

---

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
**Figure 11. Slots of the class slot-found of agathe ontology**

<table>
<thead>
<tr>
<th>Slot-Found</th>
<th>Instance*</th>
<th>Slot-Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related-Slots</td>
<td>Float</td>
<td>Instance*</td>
</tr>
<tr>
<td>Float-Values</td>
<td>Float</td>
<td></td>
</tr>
<tr>
<td>In-the-Beginning</td>
<td>Boolean</td>
<td>Instance*</td>
</tr>
<tr>
<td>Boolean-Values</td>
<td>Boolean*</td>
<td></td>
</tr>
<tr>
<td>Slots-of-Web-Page</td>
<td>Instance*</td>
<td></td>
</tr>
<tr>
<td>Region</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Start-Position</td>
<td>Integer</td>
<td>Instance*</td>
</tr>
<tr>
<td>Values</td>
<td>String</td>
<td>Instance*</td>
</tr>
<tr>
<td>Instance-Values</td>
<td>Instance*</td>
<td>Concept</td>
</tr>
<tr>
<td>Extractor-Instance</td>
<td>Instance</td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td>Instance*</td>
<td></td>
</tr>
<tr>
<td>Symbol-Values</td>
<td>Symbol*</td>
<td></td>
</tr>
<tr>
<td>Int-Values</td>
<td>Integer*</td>
<td></td>
</tr>
<tr>
<td>Slot-in-Process</td>
<td>Instance</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 12. A Web page to process**

**CEC’07 and EEE’07**

The 9th IEEE Conference on E-Commerce Technology (CEC’07) and the 4th IEEE Conference on Enterprise Computing, E-Commerce and E-Services (EEE’07) are the flagship annual conferences of the IEEE Computer Society Technical Committee on E-Commerce. In 2007, we will hold both conferences together as a joint event, providing a platform for researchers and practitioners interested in the theory and practice of E-Commerce and Enterprise Computing. The joint conference will focus on new technologies and methods that enable business processes to smoothly extend in a cross-enterprise environment, including solutions to facilitate business coalition in a flexible and dynamic manner; extending Next Generation Internet which provides ubiquitous, multimedia, and secure communication services. The program of CEC’07/EEE’07 will consist of invited talks, paper presentations, and panel discussions. We invite submissions of high quality papers describing fully developed results or ongoing work.

The theme for CEC’07/EEE’07 will be “Ubiquitous Commerce and Services on Next Generation Internet.” Topics for submission include but are not limited to:

- Commerce and trading technologies track:
  - (Track co-chairs: Rei Itaka, Hiroshima Internation Univ. and Birgit Hofreiter, Univ. of Vienna)
  - Supply chain management and e-auction technology
  - Coordination optimization in supply chain, combinatorial auctions, multi-agent negotiations, inventory management, pricing, e-Contracting, EDI
  - Marketing and advertising technology
  - SFA, CRM, recommendation systems, context aware services, user behavior modeling
  - Payment and privacy enhancing technologies
  - Payment protocols, identity management, privacy protection
- M-Commerce and P-Commerce
- Mobility management, context-dependent services, ubiquitous commerce

**IMPORTANT DATES**

- **December 1, 2006**: Workshop proposals
- **January 12, 2007**: Extended submission deadline of conference papers
- **March 31, 2007**: Notification of acceptance
- **April 30, 2007**: Camera-Ready copy of accepted papers due
- **July 23, 2007**: Workshop program
- **July 24-26, 2007**: Conference program

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
In order to test our system, we launched AGATHE using the following query: “agent, information retrieval, call for paper”. We retained the 72 first HTML pages given by the Search Subsystem. Table 1 shows AGATHE first classification results for calls for papers (workshop, conference, journal, book chapter …).

Then, we calculated the global system accuracy, which is defined as follows in Box 1.

For our pages sample, the global system accuracy was 93% of well-classified documents. We also measure recall, precision and F-measure (Van Risbergen, 1979) from our experiment. Recall is defined as follows in Box 2.

Using our system, we also obtained a recall score of 93% of relevant documents that were retrieved by the search. Precision is defined as follows in Box 3.

Using our system, we obtain a precision score of 90% of results retrieved by the search that were relevant. The F-measure is weighted as the harmonic mean of precision and recall and is defined as follows:

\[
F_\beta = \frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}
\]

We decide to use F-measure with \(\beta = 1\) in order to give same importance to precision and recall. In fact, we must retrieve all relevant documents and all retrieved documents must be relevant.

Using our system, we obtained an F-measure score of 91%. It is actually a good result, however we need to improve our system in order to have a better recognition rate and avoid false results.
Table 1. Confusion matrix

<table>
<thead>
<tr>
<th>Obtained results</th>
<th>Relevant documents</th>
<th>Irrelevant documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant documents</td>
<td>28</td>
<td>3</td>
</tr>
<tr>
<td>Irrelevant documents</td>
<td>2</td>
<td>38</td>
</tr>
</tbody>
</table>

Box 1.

Global accuracy = \[
\frac{|\{\text{relevant document}\} \cup \{\text{irrelevant document}\}|}{|\{\text{document}\}|}
\]

Box 2.

Recall = \[
\frac{|\{\text{relevant document}\} \cap \{\text{retrieved document}\}|}{|\{\text{relevant document}\}|}
\]

Box 3.

Precision = \[
\frac{|\{\text{relevant document}\} \cap \{\text{retrieved document}\}|}{|\{\text{retrieved document}\}|}
\]

RELATED WORKS

Cooperative Information Gathering proposed by Oates, Nagendra Prasad and Lesser (Oates et al., 1994), apprehends information gathering as a problem solving process distributed on cooperative agents permitting discovery and integration of relevant information clusters. Following the MACRON (Decker et al., 1995) and BIG (Lesser et al., 2000) systems and the works of Ambite and Knoblock (Ambite & Knoblock, 1997), already mentioned in the paper, agents are more and more used in the development of information retrieval and recommender systems on the Web (Lorenzi et al., 2005) (Woendl & Groh, 2005; Birukov et al., 2005). The combined use of agents and ontologies for information gathering on the Web, as proposed in AGATHE, is more recent, and the works of Cesarano, d’Acierno and Picariello (Cesarano et al., 2003) propose a system coupling agents and ontologies to perform an on-line classification of Web pages resulting of a user request on the Web, which gives a ranking more relevant than the one given by the search engine. After a preparation phase, agents reclassify the Web pages, comparing the word they contain to an ontology previously defined. This new ranking, taking into account to the semantics of the user request is more relevant.

Jung (Jung, 2007) proposes a system permitting to refine requests on the Web by automatically building and merging ontologies associated to specific areas of the Web (set of pages) with a mediator agent. Doing so, this mediator agent builds a consensual ontology which is then used by other agents to refine the requests.

(Cesarano et al., 2003) propose a system coupling agents and ontologies to perform an on-line classification of Web pages resulting of a user request on the Web, which gives a ranking more relevant than the one given by the search engine. After a preparation phase, agents reclassify the Web pages, comparing the word they contain to an ontology previously defined. This new ranking, taking into account to the semantics of the user request is more relevant.

Jung (Jung, 2007) proposes a system permitting to refine requests on the Web by automatically building and merging ontologies associated to specific areas of the Web (set of pages) with a mediator agent. Doing so, this mediator agent builds a consensual ontology which is then used by other agents to refine the requests.
CONCLUSION AND FUTURE WORK

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 article, 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 User 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 operational ontology, named Agath ontology, and/or an ontology related to the restricted domain(s) considered, to perform the tasks of classification, filtering, recommendation and/or extraction of information using reasoning.

The AGATHE system has a first complete prototype running and the first results are under production. For the improvement of this prototype, four 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, and to help minoring typical linguistic problems like polysemy, passive voice, anaphora, among other language pitfalls;

ii. the integration of learning techniques in particular for the extraction and recommendation tasks; the inclusion of these techniques could accelerate knowledge acquisition and thus the production of a solution to a new domain;

iii. support tools that help design solutions for new domains; for instance, a rule designer which creates according to parameters provided by the users is already available; and finally

iv. 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 (2006), could be used in a forthcoming version of AGATHE.

Solutions for the first three axes are currently already under development as a result of the settled cooperation between the two research groups from Brazil and France.

REFERENCES


Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Bernard Espinasse is currently full professor in computer sciences at the Ecole Polytechnique Universitaire of Aix-Marseilles University, and team leader at the LSIS UMR CNRS research laboratory, Marseilles, France. He has been associate professor at the Laval University of Québec, Canada (1983-1987). He received an engineer diploma from the Ecole Nationale Supérieure d’Arts et Metiers (1977), and a PhD of the Aix-Marseilles University (1981). His main research topics are software agents, agents based simulation, ontologies and semantic Web, decision support and information systems. He is author or co-author of numerous books and publications in international journals and conferences on these topics, and member of several editorial boards, and of numerous international program committees.

Sébastien Fournier is at present associate professor at the Aix-Marseilles University and researcher at the LSIS UMR CNRS research laboratory, Marseilles, France. He holds a master of sciences at the Pierre and Marie Curie University in Paris (2001), and a PhD from the Naval Academy Research Institut, Brest (2005). His research interests focus on multiagent systems, ontologies and semantic Web, information retrieval and gathering, geographic information systems, qualitative spatial reasoning and simulation.

Frederico Freitas is currently adjunct professor at the Federal University of Pernambuco, at Recife, Pernambuco, Brazil. He graduated in informatics at the Aeronautics Technological Institute (ITA), and received his PhD in electrical engineering from the Federal University of Santa Catarina (2002). He researched for almost a year at the Department of Informatics of the University of Karlsruhe, as a member of the Brazilian-German project “A semantic approach to data retrieval”. He has co-chaired two workshop series, WONTO (Workshop on Ontologies and their applications) in Brazil, and BAOSW (Building Applications with Ontologies for the semantic Web) in Portugal. He also co-edited special issues on related themes at JBCS (Journal of Brazilian Computer Science) and JUCS (Journal of Universal Computer Science). His interest areas comprise ontologies, semantic web, multi-agent systems, knowledge representation and text mining. He is author or co-author of some publications in journals and international conferences on these topics, and member of several international program committees.