E-Business Applications for Product Development and Competitive Growth:
Emerging Technologies

In Lee
Western Illinois University, USA

BUSINESS SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content: Kristin Klinger
Director of Book Publications: Julia Mosemann
Acquisitions Editor: Lindsay Johnston
Development Editor: Myla Harty
Publishing Assistant: Julia Mosemann
Typesetter: Michael Brehm
Production Editor: Jamie Snavely
Cover Design: Lisa Tosheff

Published in the United States of America by
Business Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

Library of Congress Cataloging-in-Publication Data

E-business applications for product development and competitive growth :
emerging technologies / In Lee, editor.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-60960-132-4 (hbk.) -- ISBN 978-1-60960-134-8 (ebook) 1.
Electronic commerce. 2. Information technology--Management. 3. New products.
4. Technological innovations--Management. I. Lee, In, 1958-
HF5548.32.E17369 2011
658.5'7502854678--dc22
2010046732

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 12
AGATHE-2:
An Adaptive, Ontology-Based Information Gathering Multi-Agent System for Restricted Web Domains

Bernard Espinasse
Aix-Marseilles University, France

Sébastien Fournier
Aix-Marseilles University, France

Fred Freitas
Universidade Federal de Pernambuco, Brazil

Shereen Albitar
Aix-Marseilles University, France

Rinaldo Lima
Universidade Federal de Pernambuco, 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, with the use of domain ontology, 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 can be instantiated in information gathering systems for specific domains, such as academia, tourism, etc. In this chapter, the authors present this approach, a generic software architecture, named AGATHE-2, which is a full-fledged scalable multi-agent system. Besides offering an in-depth treatment for these domains due to the use of domain ontology, this new version uses machine learning techniques over linguistic information in order to accelerate the knowledge acquisition necessary for the task of information extraction over the Web pages. AGATHE-2 is an agent and ontology-based system that collects and classifies relevant Web pages about

DOI: 10.4018/978-1-60960-132-4.ch012

a restricted domain, using the BWI (Boosted Wrapper Induction), a machine-learning algorithm, to perform adaptive information extraction.

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, Shaw, & Simon, 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 (Mc Callum 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, Minton, & C. Knoblock, 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 (Kushmerick, 1999a). 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.

On the other hand, it is widely known that Machine Learning (ML) algorithms simplify the development of IE programs; these algorithms have been utilized to automate extraction rules' production. In recent times, many IE systems had been developed following a three-step procedure: (1) Recognizing relevant information in the text (2) Extracting this information (3) Storing it in an organized structure or in a database (Kushmerick, 1999b; Siefkes & Siniakov, 2005).

In the last years, we have conducted research with these research hypotheses, and produced ontology-based restricted-domain cooperative information gathering software agents accordingly, that permit the development of a specific information gathering systems e.g. the MASTER-

Web system (Freitas & Bittencourt, 2003), and a first version of AGATHE system (Espinasse et al, 2008). According to this approach and based on previously-presented guiding ideas, this chapter presents a generic software architecture, named AGATHE-2, an extension of AGATHE system (Espinasse et al., 2008) that permits a more adaptive information gathering on restricted Web domains. As its predecessors, AGATHE-2 is an agent and ontology-based system that collects and classifies relevant Web pages from a restricted Web domain. Furthermore, it uses the BWI (Boosted Wrapper Induction) [ref], a machine-learning algorithm, to perform adaptive information extraction over the collected Web pages.

The chapter is organized as follows: section 2 introduces major notions 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 presents the AGATHE-2 system, a multi-agent architecture for the development of intelligent gathering systems on the Web, its objectives, its architecture with its three main subsystems, and its general functioning. Sections 4 to 6 present in detail these subsystems composing AGATHE-2 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, ONTOLOGIES AND MACHINE LEARNING

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, 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 & 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 et al, 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 sub-goals within those plans. MACRON is composed of three types of autonomous agents: reasoning agents, low-level retrieval agents, and user interface agents.

(Lesser et al., 2000) have proposed the BIG system. This system 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.

Motivation of Using 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 their 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 Types 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:

1. *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.
2. *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 (Kushmerick, 1999a) fail. Wordnet (Miller et al, 1990) and the ontologies from the GATE project (General Architecture for Text Extraction) are good examples of ontologies of this type.
3. *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.

Machine Learning for Information Extraction: Adaptive Information Extraction

Rule-based IE systems, as AGATHE, used to be developed in an ad hoc manner; however the process of knowledge engineering required to come up with the rules is laborious and time consuming. Domain specific extraction rules or patterns, instead of being handcrafted for each application domain, can be learnt directly by generic domain-independent IE system using a tagged domain-specific training set (Etzioni et al, 2008). Enriched with natural language processing (NLP) and inductive logic programming (ILP) methods, a spectrum of generic architectures were proposed to realize supervised IE on Web pages (Turmo et al, 2006).

Recently, self-supervised IE systems, a new paradigm in the IE research domain, has come to life. In general, these systems depend on domain-independent patterns to label their training sets for each application domain (Etzioni et al., 2008). According to the under test performance of self-supervised systems, supervised IE systems are still more attractive. In order to evaluate and compare IE supervised algorithms, three measures are principally used: Precision, Recall and F-Measure (Maynard et al, 2006).

As for the task of text categorization, in the last decade, different classifications for adaptive information extraction methods were proposed (Siefkes & Siniakov, 2005; Tang et al 2007). According to the classification proposed in (Tang et al., 2007), the three distinguished classes of adaptative IE methods are: Rule Learning based methods, Classification based Methods, and Sequential Labeling based Methods.

AGATHE-2 has adopted the Boosted Wrapper Induction (BWI) algorithm (Freitag & Kushmerick, 2000; Kauchak et al 2004), a Rule Learning based method belonging to the Wrapper Induction category. The reason for which BWI was chosen for this work is its competence in information extraction from unstructured text in addition to structured and semi-structured text (Albitar et al., 2010; Lima et al, 2010)

AGATHE OVERVIEW

The AGATHE project is an international cooperation between France and Brazil. Its aim is proposing a generic software architecture that allows the development of information gathering systems on the Web, for one or for a few restricted domains. In the AGATHE architecture, 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, as any multi-agent-based solution.

AGATHE software architecture benefits from agent-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 the MASTER-Web system (Freitas & Bittencourt, 2003), which uses ontologies to carry out the tasks of classification and extraction of information on the Web on only one restricted domain of search at a time. AGATHE is enabled not only to perform over more than one domain at a time, but also to establish cooperations among different domains. For instance, a cooperation between the domains of Academia and tourism could be held in the following way: 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, touristical visits, etc.).

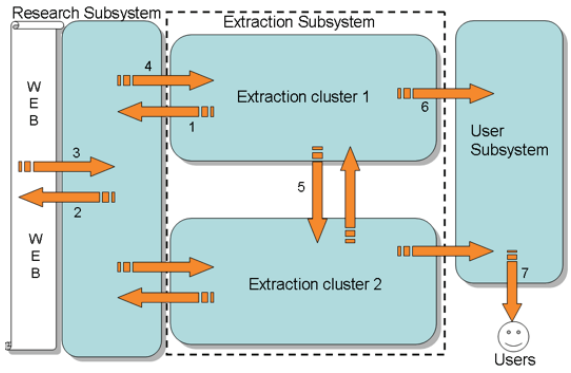
AGATHE-2, as its previous version AGATHE (Espinasse et al., 2008) is an extension of MASTER-Web, thus reusing the latter's ontologies (Frame ontologies) and deploys their use onto a complex and distributed organization of more effective software agents, with different types of specialized agents in interaction. 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 implementation, AGATHE respects the recommendations defined by the FIPA ("FIPA," 2000).

As symbolic rules were domain dependent and arduously written, in particular for the information extraction task from collected Web pages, it seemed judicious to replace them by machine supervised learning techniques. Thus, in order to make more adaptive the AGATHE system, recent works relative to the use of machine learning techniques (and particularly the efficient and effective BWI algorithm [Kauchak et al 2004]) for information extraction tasks from Web pages, have lead to a new version of this system, a more adaptive version named AGATHE-2.

The main objective of the whole AGATHE project is to accomplish 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). With each of these search domains there is a specific ontology associated.

In this section, the general architecture and functioning of AGATHE-2 system are firstly presented, then the three main subsystems composing AGATHE-2 system are introduced. Finally, some implementation details are given.

Figure 1. General AGATHE architecture



Architecture and General Functioning of AGATHE-2 System

The AGATHE-2 general architecture, 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)*.

The three main subsystems of AGATHE-2 are themselves multi-agent systems (MAS), 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-2 system 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.

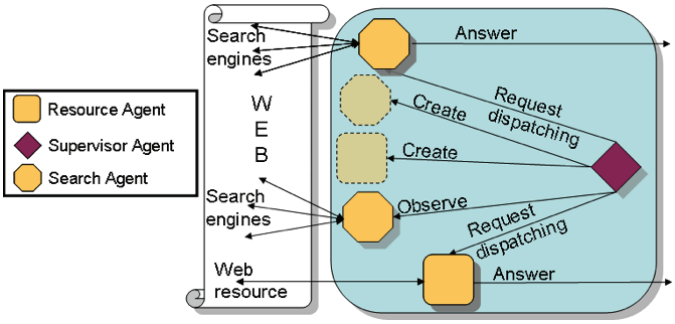
For the construction and the handling of ontologies, defined by Frames, the Protégé environment (Protégé, 2006) is used, and the exploitation of the ontologies by the Jess agents is done via the Protégé plugin JessTab (Eriksson, 2003).

The information extraction task, performed by the ESS subsystem, uses the WEPAIES system (Lima et al., 2010). This system integrates various software modules to clean, prepare, annotate and produce, by supervised learning, information extractors used in AGATHE-2. In order to clean the Web pages it uses the HTMLCleaner tool (Girardi, 2007). The Web pages, are annotated with POS (Part Of Speech) tags, by employing the QTAG tool (Tufis & Mason, 1998). Finally, capitalizing over the linguistic patterns, an automatic extractor based on extraction rules is produced using a specially tailored version of TIES system("TIES," 2004) developed by our group (Lima et al., 2010).

The classification results and information extracted are stored in a MySQL relational management database system according the RDF Format and use Jena framework ("Jena," 2006). 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. Jena framework is used in AGATHE-2 to store in the database the classification results, and extracted information in RDF format. These tasks are performed by the Storage Agent of the Extraction Subsystem. Jena framework is also used to exploit this database in the User Subsystem.

The following section presents in detail the three subsystems composing AGATHE-2 system.

Figure 2. Internal architecture of the Search Subsystem



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 that 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, and\or others). It receives requests from a 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 of AGATHE-2, the different ontologies used in this Subsystem to perform classification and extraction tasks, the way that agents employ these ontolo-

gies, and finally, we present in details each agent of this subsystem.

Extraction Subsystem Architecture

The AGATHE-2 system have 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: first to ask for Web pages to the Search Subsystem, and then 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 then whole system, and consists in the subtasks of page validation, functional classification, and information extraction.

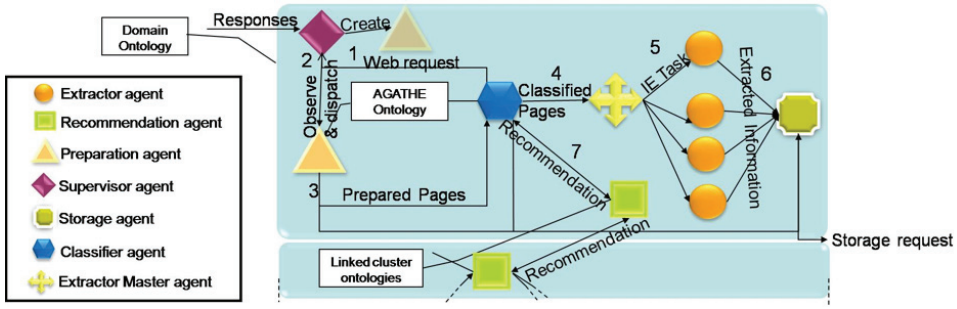
The AGATHE-2 extraction subsystem is quite different from the AGATHE extraction subsystem, because it performs an adaptive information extraction based on supervised learning techniques. This new subsystem is composed of extraction clusters which are, in turn, composed of software agents that performs pages classification and information extraction.

Software agents using symbolic rules that exploit ontologies used to realize most tasks of information classification and extraction in the original AGATHE system (Espinasse et al., 2008). Since symbolic rules were domain dependent and had to be written manually at a high human-labour cost machine learning techniques seemed to us welcome to help develop extractors automatically in a faster way. Therefore, the aim of our recent work was to combine, in AGATHE-2, symbolic based classification and machine learning based information extraction in new extraction subsystem architecture.

Agents exploit a domain ontology using symbolic rules in order to realize the tasks of classification. For the information extraction task, for each relevant web page class of this ontology, an specific agent extracts information according to extractors obtained by a supervised learning phase realized by BWI algorithm (Albitar et al., 2010; Lima et al., 2010).

The new extraction cluster is presented in Figure 3. Every extractor agent wraps a running WEPAIES for IE, while the classifier agent realizes semantic classification only. An extractor master agent is introduced in this cluster in order

Figure 3. Internal architecture of an Extraction Cluster



to ensure system modularity and agent specialization. This agent manages the allocation of IE tasks to extractor agents according to a predetermined strategy. Finally, the storage agent stores extracted information in a database for future analysis

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. These agents perform specific tasks in the extraction cluster. Some of these agents use ontologies to perform their tasks. The various agents that compose the cluster are:

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

The functioning of the Extraction Cluster is the following (Figure 3):

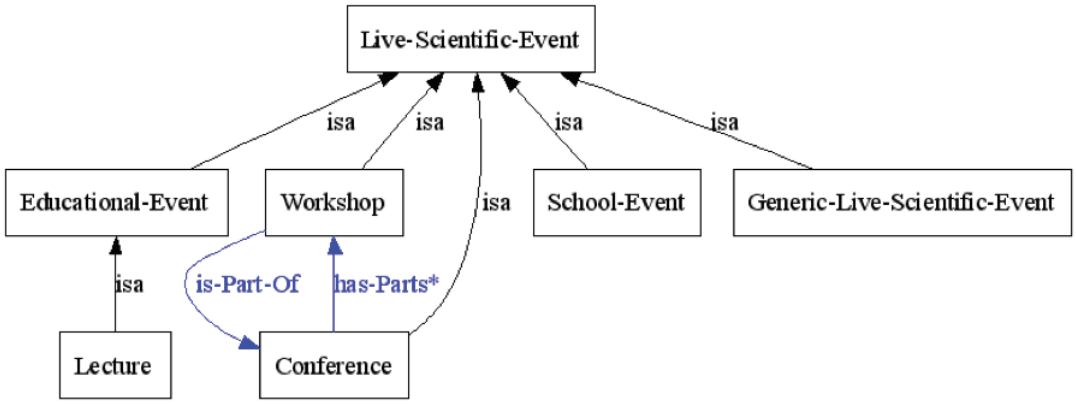
1. The classifier agent sends a demand for a web search to the supervisor agent.
2. The supervisor agent forwards the demand towards the search subsystem and then sends the retrieved pages to the preparation agent.
3. After filtering and functionally classifying (discarding emails and lists), preparation agent sends valid web pages to the classifier agent and sends preparation results to the storage agent.
4. Web pages belonging to the cluster's specific domain are sent by the classifier agent, after classifying them semantically, to the extractor master agent.
5. The extractor master agent dispatches IE task between multiple extractor agents according to a predetermined distribution strategy.
6. Each extractor agent realizes the assigned IE task and sends extracted information to the storage agent in order to be stored in the database.
7. If the classifier agent discovers that the web page doesn't belong to its domain, it forwards it to the recommendation agent who decides to which cluster the page must be sent.

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 AGATHE Extraction Subsystem exploit two types of ontology to perform tasks of pages classification and

Figure 4. A part of the "Science ontology" concerning scientific events



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 4 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 Agathe ontology is already defined and used in MASTER-Web system and named inside "Web ontology". This ontology specifies the main concepts used by AGATHE-2 for the classification and extraction tasks performed on Web pages. Figure 5 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).

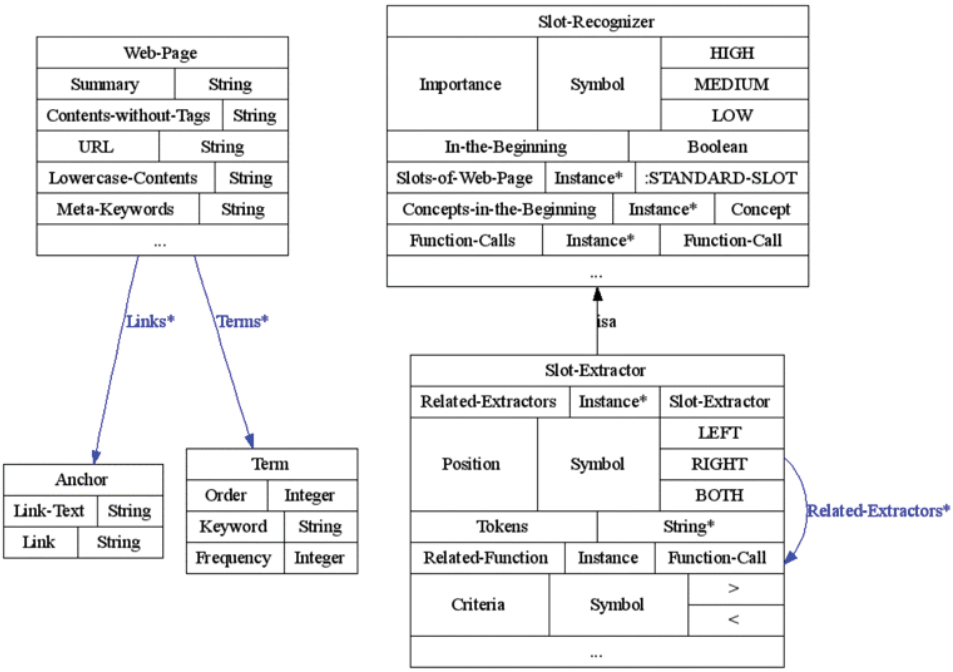
How Agents Use Ontologies

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:

Figure 5. Main classes, slots and relations of the operational ontology



This specific rule permits to find interesting data in text associated to links that help to classify a page. The name of this rule is "r_454" and its purpose is to identify if a certain slot (?s, in the rule) is identified and it uses the Processing-Monitor and Slot-Recognizer concepts that are a part of the Agathe ontology (see Figure 5). 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 ($?cb, in the rule) in the text of a Web link and whether the absent concepts are not present ($?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

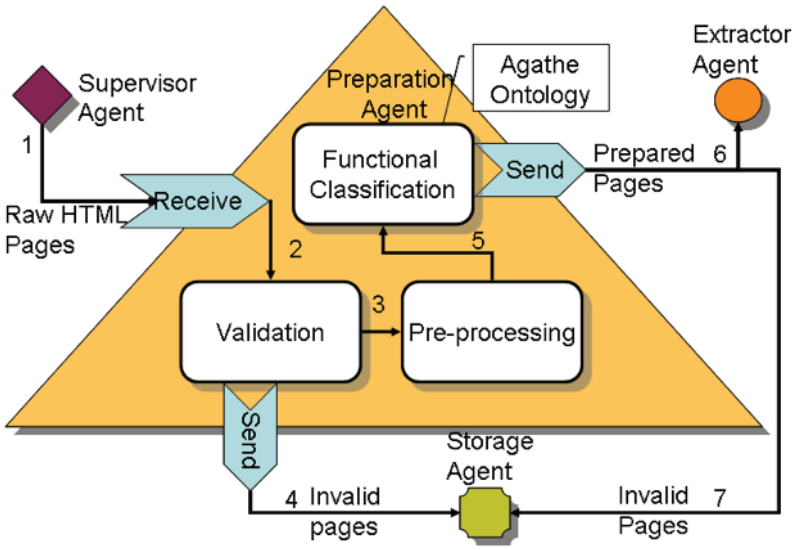
Preparation Agents receive Web pages from the Search Subsystem and perform some treatment on them that will be described below. These agents are created by the Supervisor Agent of the considered extraction cluster and are deleted by this same agent when they are not being used any more. These agents 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 below (figure 6):

- *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,

Algorithm 1.

```
(defrule r_454
(object (Page-Status STORED) (is-a Processing-Monitor))
?f <- (object (Importance MEDIUM) (is-a Slot-Recognizer)
(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] Val-
ues)) 0)
(> (count-occurs-once (words-of-concepts $?cb) (slot-get [LINKS-TEXTS] Val-
ues)) 0)))
=>
(slot-insert$ [SLOTS-FOUND] Instance-Values  ?s))
```

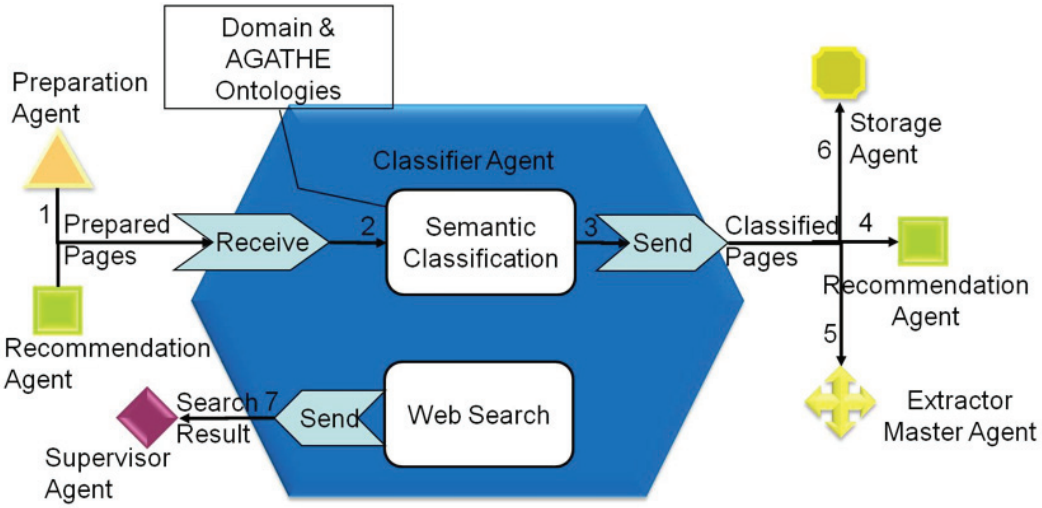
Figure 6. A Preparation Agent and its internal tasks



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

Figure 7. The Classifier Agent and its internal tasks



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.

In order to achieve better performance and flexibility, it is possible to duplicate such preparation agents to reduce their strain.

The Classifier Agent

As defined in previous version of AGATHE, this agent classifies Web pages semantically, using symbolic Jess rules and exploiting the domain ontology; if the page belongs to a relevant class of the domain, it sends it to the extractor master agent, otherwise it sends it to the recommendation agent of its cluster. Moreover, the page's class (ex. Conference, workshop, journal, etc.), with its address as well as other details are sent by this agent to the storage agent to be eventually stored in the database. The Figure 7 presents the Classifier Agent and its internal tasks.

The Extractor Master Agent

Web pages belonging to a specific class of the considered domain might contain information related to different concepts in the domain ontology. While AGATHE has a multi-agent architecture, it appeared coherent and beneficial to distribute the IE load among multiple extractor agents according to user-defined strategies. Consequently, it was legitimate to introduce the extractor master agent to dispatch IE tasks: first, it receives a classified web page, then it resends it toward the extractor agents specialized in its class according to the defined distribution strategy. The Figure 8 presents the Extractor Master Agent and its internal tasks.

Figure 9 illustrates the class based distribution strategy actually adopted in AGATHE. Each extractor agent is specialized in extracting information from pages belonging to a specific class in the scientific domain.

The Extractor Agents

The Extraction Cluster is composed of several Extractor Agents associated to a specific domain ontology linked to the cluster. The task of these agents is to perform an adaptive information

Figure 8. The Extractor Master Agent and its internal tasks

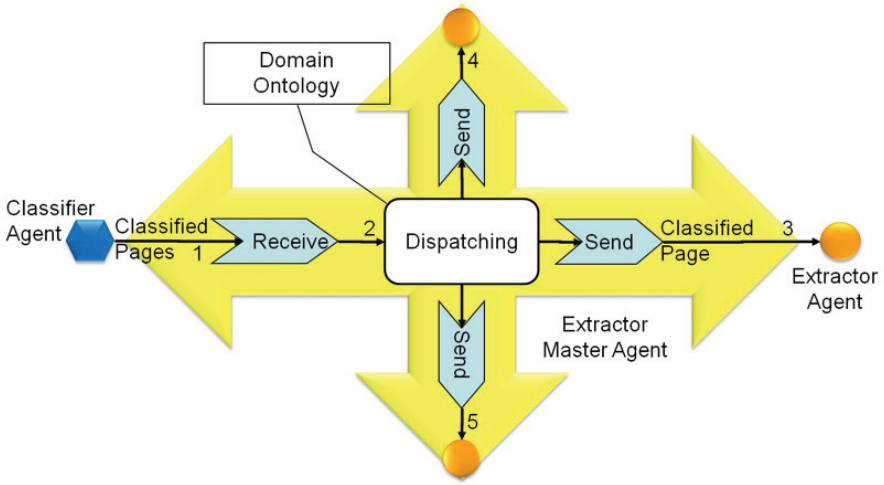
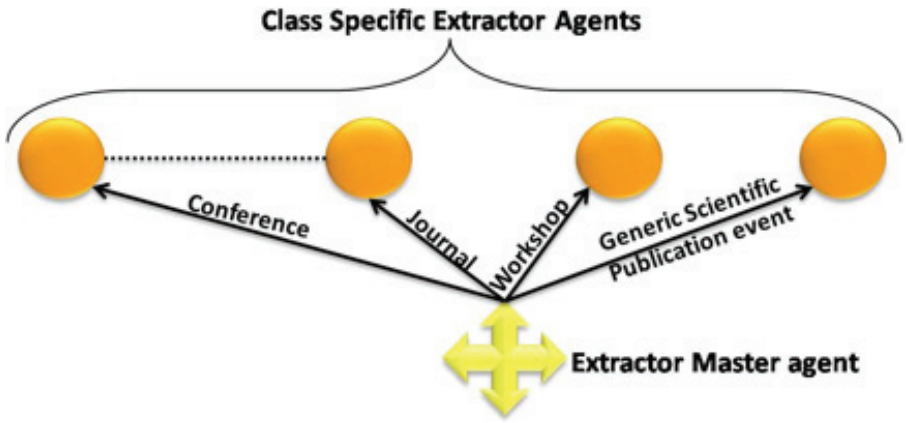


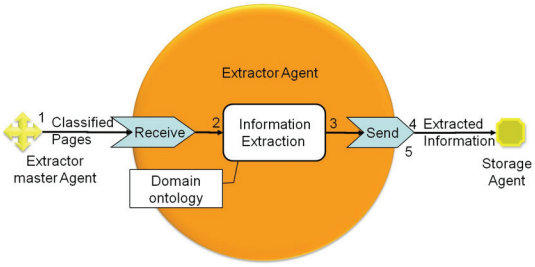
Figure 9. Distribution strategy in the extraction subsystem



extraction on these pages by applying extractors obtained by supervised learning. This supervised learning is done using a training set of relevant annotated Web pages of the concerned domain. Corpus annotation, which can be done manually by domain experts, uses domain ontology's concepts, and might be completed automatically by POS (Part of Speech) tagging, that takes into account the morphosyntactic structure of the natural language. Each Extractor Agent is associated to a particular class (concept) of the domain ontology to extract.

This agent wraps WEPAIES (Lima et al, 2010), an IE system implementing BWI which is a supervised machine learning algorithm. It extracts relevant information from web pages by executing WEPAIES, using wrappers produced off-line during a training step and saved in XML files. Choosing a concept-based distribution strategy, a single domain concept is delegated to each extractor. Consequently, as soon as an extractor agent receives a message, it runs WEPAIES to extract specific concepts from the page specified in the message. Wrapped inside AGATHE,

Figure 10. An Extraction agent and its internal tasks



WEPAIES is always executed in "IE" mode; supervised training is carried out independently. The Figure 10 presents the Extractor Agent and its internal tasks.

Depending on the classification results for a Web page, and making use of ontologies, the Extractor agent performs the information extraction task. 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 IE task according to the assigned class, the extracted information is then transmitted to the Storage Agent.

The Recommendation Agent

The Recommendation Agent (Figure 11) 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:

- *Inter-Domain recommendation*: it recommends pages/links that might have some interest to other Classifier Agents of the cluster.
- *Intra-Domain recommendation*: it recommends some pages/links to other Extraction Clusters, pages that could be interesting for them. For this task, the Recommendation Agent needs to access ontologies from the different clusters involved. It dispatches these pages to the various Recommendation Agents of the cluster concerned. An example is briefly described below.
- *Dispatching*: it forwards 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. It is typically available in these pages under the label "social program".

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 Agent stores in persistent memory the instances of the class "Slot-Found" of the Agathe ontology, class where the

Figure 11. Recommendation agent and its internal tasks

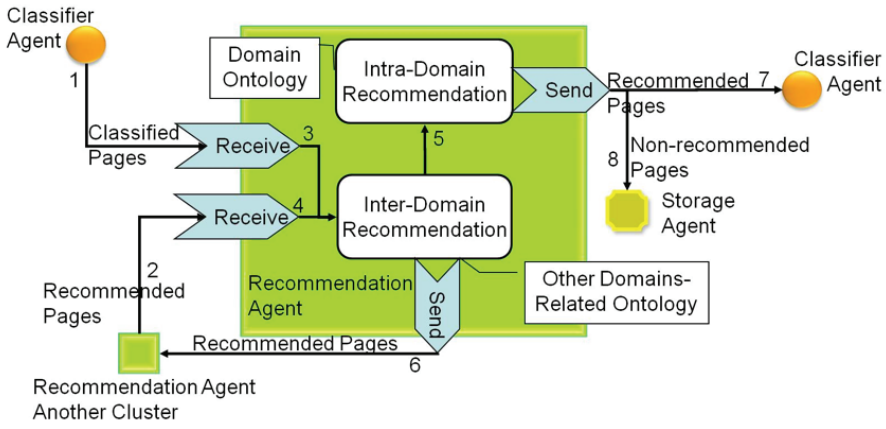
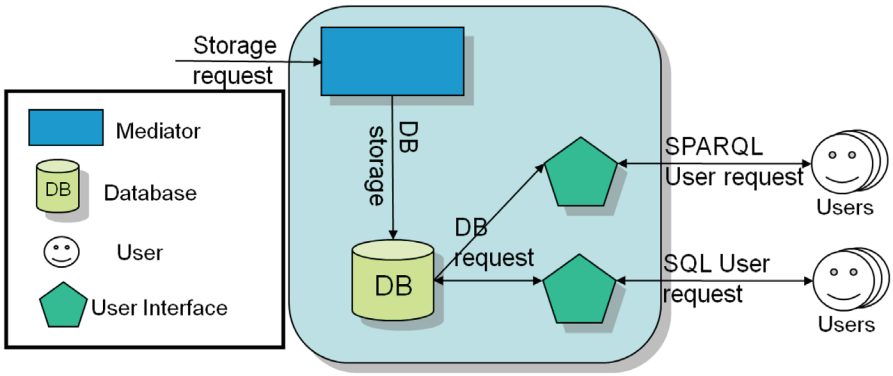


Figure 12. Storage agent



extracted information is stored by the Extraction Agent, and store these information in the RDF format in a relational data base. 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 12).

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

Table 1. Frequencies' concepts in the CFP test corpus

Concepts to extract	Frequencies in the corpus			
	Learning	%	Test	%
workshopname	543	11.8	245	10.8
Workshopacronym	566	12.3	243	10.7
Workshophomepage	367	8.0	215	9.5
Workshoplocation	457	10.0	224	9.9
workshopdate	586	12.8	326	14.3
workshopsubmissiondate	590	12.9	316	13.9
Workshopnotificationacceptancedate	391	8.5	190	8.4
Workshopcamerareadycopydate	355	7.7	163	7.2
conferencename	204	4.5	90	4.0
Conferenceacronym	420	9.2	187	8.2
conferencehomepage	104	2.3	75	3.3
TOTAL	**4583**	**100**	**2274**	**100**

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 Papers (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.

Experimental Conditions

In order to evaluate the AGATHE-2 system's performances, we used in our experiments the Call For Papers (CFP) test corpus of the Pascal Challenge (Pascal Challenge, 2005) (200 pages containing 2274 annotated fields) without taking into consideration text annotation. This corpus had been used earlier as a formal basis to evaluate and compare the performance of different ML algorithms (Ireson et al., 2005). The majority of the 1100 documents (850 calls for workshops, 250 calls for conferences) constituting the CFP corpus come from the domain of computer science. Others were collected from the biomedicine and linguistics domains.

The 1100 documents were arranged in three different corpuses; the training corpus containing 400 calls for workshops, the test corpus containing 200 calls for workshops and the enrich corpus containing 250 calls for workshops and 250 calls for conferences. A call for workshop might include details regarding conferences as workshops might be related to other conferences.

In the corpus' pages, 11 information to extract corresponding to concepts of the domain ontology, are annotated: 8 for workshop class concepts and 3 for conference class concepts. Some concepts like dates can be shared between both classes. Conference concepts have lower frequencies in corpus compared to workshop concepts. Table 1 shows the 11 concepts' frequencies in both training and test corpuses.

Table 2. Classification results

Number of pages	Classified as conference	Classified as Workshop	Unclassified pages
199	28	169	2

Classification Results

According to the first version of AGATHE (Espinasse et al., 2008), classification task based on symbolic rules resulted in good precision and recall. For this reason, we maintained the symbolic approach based part for the semantic classification.

Statistical results presented in table 2 shows that AGATHE had some difficulties in classifying some test corpus' pages (200 calls for workshops) as both classes (workshop and conference) are very close. Anyway, AGATHE was able to classify most corpus pages as workshop as it obtained (86% and 85%) rates for the precision and the recall respectively.

Information Extraction

Here we demonstrate how AGATHE works taking as example the call for paper of the conference RCIS2010 presented in Figure 14.

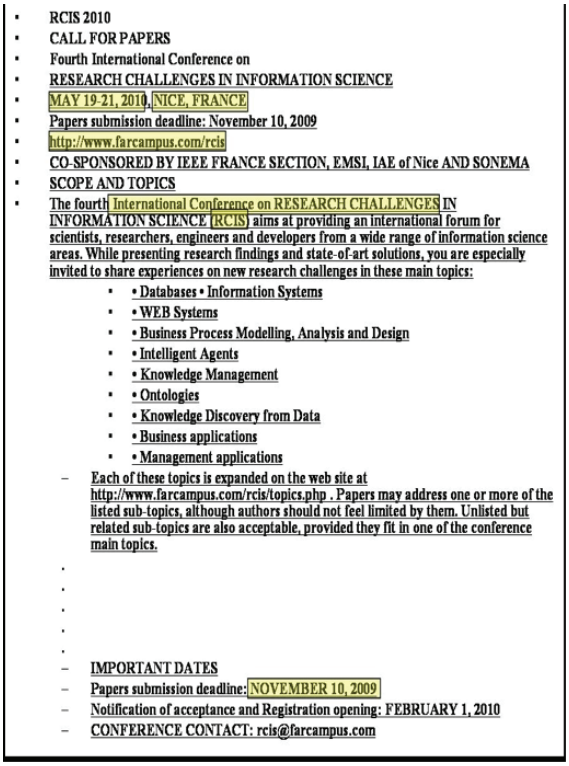
First, the page was prepared and then classified as conference that is the correct class of the page. Then, the extractor agent extracted information from the page. The resulting information, highlighted in Figure 13, is presented in table 3. As workshop and conference classes are so close, they share many concepts like date, location etc. This was the case of last five extracted concepts.

Information Extraction Results

AGATHE (Espinasse et al., 2008) didn't deliver promising results concerning the IE task. This task was implemented in symbolic rules, which were costly to develop. This was the reason for which we adopted an adaptative IE algorithm in the new version that demonstrated considerably promising results.

After statistical analysis for about 10000 database entries retrieved from treating the test corpus, the average value of F-measure for work-

Figure 13. IEEE RCIS 2010 CFP



RCIS 2010
CALL FOR PAPERS
Fourth International Conference on
RESEARCH CHALLENGES IN INFORMATION SCIENCE
MAY 19-21, 2010, NICE, FRANCE
Papers submission deadline: November 10, 2009
http://www.farcampus.com/rcis
CO-SPONSORED BY IEEE FRANCE SECTION, EMSI, IAE of Nice AND SONEMA
SCOPE AND TOPICS
The fourth International Conference on RESEARCH CHALLENGES IN INFORMATION SCIENCE (RCIS) aims at providing an international forum for scientists, researchers, engineers and developers from a wide range of information science areas. While presenting research findings and state-of-art solutions, you are especially invited to share experiences on new research challenges in these main topics:
• Databases • Information Systems
• WEB Systems
• Business Process Modelling, Analysis and Design
• Intelligent Agents
• Knowledge Management
• Ontologies
• Knowledge Discovery from Data
• Business applications
• Management applications
– Each of these topics is expanded on the web site at http://www.farcampus.com/rcis/topics.php . Papers may address one or more of the listed sub-topics, although authors should not feel limited by them. Unlisted but related sub-topics are also acceptable, provided they fit in one of the conference main topics.
– IMPORTANT DATES
– Papers submission deadline: NOVEMBER 10, 2009
– Notification of acceptance and Registration opening: FEBRUARY 1, 2010
– CONFERENCE CONTACT: rcis@farcampus.com

Figure 14. F-measure of extracted concepts in Pascal corpus

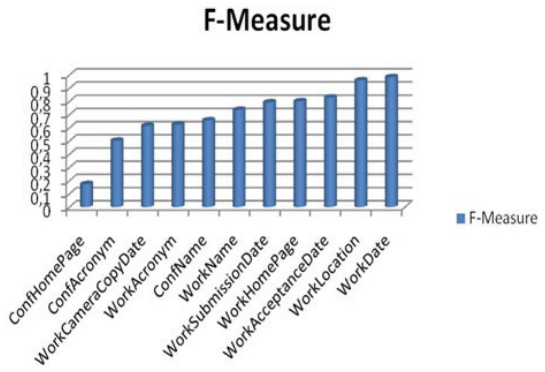


Table 3. Extracted information for IEEE RCIS 2010 CFP

Concept	Value	Start	End
Conferencename	International Conference on RESEARCH CHALLENGES	304	351
Conferenceacronym	RCIS)	376	381
Workshopdate	MAY 19 - 21, 2010	104	119
Workshophomepage	deadline: November 10, 2009 http:/ / www. farcampus. com / rcis	152	209
Workshoplocation	NICE, FRANCE	121	133
Workshoppapersubmissiondate	deadline: NOVEMBER 10	4488	4509
Workshopnotificationofacceptancedate	NOVEMBER 10, 2009	4498	4515

shop concepts was equal to (70%). Furthermore, for some concepts like *workshoplocation* and *workshopdate*, highest values (more than 85%) were observed for both precision and recall. Figure 13 illustrates F-measure values obtained during our experimentation concerning different domain concepts (Albitar et al., 2010). Lowest F-measure values were obtained for *conferenceacronym* and *conferencehomepage* concepts. Such low values come from the low frequency of these concepts in the training corpus; as it does not contain enough instances of both concepts so WEPAIES was not trained well to extract them (see Table 1).

Discussion

As a training corpus is used to learn how to extract information, similar information contexts are expected in pages to extract otherwise our system won't be able to extract target fields. This was the case for the concept *workshopnotificationofacceptancedate* that AGATHE-2 could not extract well. The reason to this error is that three words "and registration opening" were placed after "notification of acceptance" taking the place expected for the acceptance date as it was learned and registered in detectors' patterns. This kind of error is explainable as training corpus might not cover all possible cases.

Other sources of confusion might be in the length and the position of field. This was the case for the *conferencename* that was not entirely extracted. Nevertheless, this kind of confusion might be resolved as AGATHE-2 permits user intervention to decide on the correct result.

In (Lima et al., 2010) an investigation was conducted in the direction of using the BWI algorithm as an IE method for unstructured natural language documents. Table 4 shows the results obtained using the WEPAEIS on 3 corpora: the Pascal challenge Call for Papers (CFP), Seminars, and Job (http://www.isi.edu/info-agents/RISE/repository.html). These corpora were chosen because they present decreasing levels of structuring, from the more structured corpus (Seminars) to the less structured one (CFP). The results are very slightly better than the original BWI algorithm without POS tagging.

Concerning the influence of the POS tagging in information extraction process with BWI algorithm, we note that they obtained the better results for the CFP Pascal Challenge Corpus. On the other hand, for the other corpora (Seminars and Jobs) there was practically no difference. These results were justified by analyzing the more structured nature of the documents in these corpora in which the inducted wrappers could achieve very good performance without POS annotation.

Additionally, (Lima et al. 2010) have also shown that the gain in performance (F-measure) for some concepts was more the 5%, which permitted them to conclude that the use of POS tagging combined with BWI pays off against highly unstructured texts.

Table 4. Influence of the POS tagging on 3 corpora information extraction

Corpus	P	R	F1	P	R	F1
Seminars	0.974	0.953	0.963	0.971	0.964	0.967
Jobs	0.945	0.778	0.853	0.939	0.780	0.853
CFP	0.891	0.571	0.696	0.896	0.591	0.712
	(a) no POS			(b) with POS		

RELATED WORK

Cooperative Information Gathering proposed by (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 (1997), agents are being more and more used in the development of information retrieval and recommender systems on the Web (Birukov et al, 2005; Lorenzi et al, 2005; Woerndl & Groh, 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 et allii (2003) and Jung (2007) are in this trend.

(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 (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.

Other information gathering systems have also adopted this approach, particularly the CROSSMARC system (Karkaletsis et al., 2003) that was implemented for e-retail and job offers domains coupling symbolic rules with wrapper induction.

CONCLUSION AND FUTURE WORK

While being limited to restricted domains, our research hypothesis is that taking into account contexts leads to more relevant information gathering. In the last years, we have concentrated in these research issues and produced ontology-based restricted-domain cooperative information gathering software agents accordingly, that permit the development of a specific information gathering systems e.g. the MASTER-Web system (Freitas & Bittencourt, 2003), and a first version of AGATHE system, AGATHE (Espinasse et al., 2008). Furthermore, in the AGATHE-2 system we have employed machine learning techniques that simplified the development of IE; these algorithms have been utilized to automate rule production for extracting data, endowing with speed the instantiation of a solution for a new domain to be dealt by the system.

The use of supervised ML techniques in IE, Boosted Wrapper Induction (BWI), in information gathering on restricted web domains, improved the task of information extraction in terms of portability and performance. Indeed, the results obtained from this new version of AGATHE-2 system, combining agent and wrapper induc-

tion in its information extraction task, are very encouraging.

In a short-term perspective, for the IE task, we intend first to reduce its time, especially by using efficient distribution strategies to dispatch this task. Multiple class specialized extractor agents might be deployed in parallel distributing the information extraction load among them.

In a middle-term perspective, in order to improve the relevance of results, we intend to improve the classification task in particular by using pre and post treatments exploiting ontologies. The integration of techniques of natural languages processing in this task, could improve this SC task, by minoring typical linguistic problems like polysemy, passive voice, anaphora, among other language pitfalls. Then integration of learning techniques in this task could also accelerate knowledge acquisition and increase the adaptivity of the system

In more long-term perspective, we intend to use 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 (The Satine Project, 2006), could be used in a forthcoming version of AGATHE.

REFERENCES

Albitar, S., Espinasse, B., & Fournier, S. (2010). Combining Agents and Wrapper Induction for Information Gathering on Restricted Web Domains. *In Proceedings of the fourth international conference on research challenges in information science*.

Ambite, J. L., & Knoblock, C. A. (1997). Agents for Information Gathering. *IEEE Expert: Intelligent Systems and Their Applications, 12*(5), 2-4. doi:http://dx.doi.org/10.1109/64.621219

Birukov, E., Blanzieri, E., & Giorgini, P. (2005). *Implicit: A Recommender System that uses Implicit Knowledge to Produce Suggestions.* In Proceedings of the Workshop on Multi-Agent Information Retrieval and Recommender Systems at the 19th International Joint Conference on Artifical Intelligence (IJCAI-05)

Cesarano, C., d'Acierno, A., & Picariello, A. (2003). An intelligent search agent system for semantic information retrieval on the internet. In *WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management* (p. 111–117). New YorkACM. doi:http://doi.acm.org.gate6.inist.fr/10.1145/956699.956725

Decker, K., Lesser, V., Prasad, M. V. N., & Wagner, T. (1995). MACRON: An Architecture for Multi-agent Cooperative Information Gathering. *In Proccedings of the CIKM-95 Workshop on Intelligent Information Agents*.

Eriksson, H. (2003). Using JessTab to Integrate Protégé and Jess. *IEEE Intelligent Systems, 18*(2), 43–50. doi:http://dx.doi.org.gate6.inist.fr/10.1109/MIS.2003.1193656

Espinasse, B., Fournier, S., & Freitas, F. (2008). Agent and ontology based information gathering on restricted web domains with AGATHE. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing* (p. 2381–2386). New York: ACM. doi:http://doi.acm.org.gate6.inist.fr/10.1145/1363686.1364252

Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2008). Open information extraction from the web. *Commun. ACM, 51*(12), 68-74. doi:http://doi.acm.org/10.1145/1409360.1409378

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., et al. (2004). Web-scale information extraction in knowitall: (preliminary results). In *WWW '04: Proceedings of the 13th international conference on World Wide Web* (p. 100–110). New York: ACM. doi:http://doi.acm.org.gate6.inist.fr/10.1145/988672.988687

FIPA. (2000). *The Foundation for Intelligent Physical Agents*. Retrouvé Juillet 13, 2010, from http://www.fipa.org/

Freitag, D., & Kushmerick, N. (2000). Boosted Wrapper Induction. Proceedings of the 17th National Conference on AI and 12th Conference on Innovative Applications of AI.

Freitas, F., & Bittencourt, G. (2003). An Ontology-Based Architecture for Cooperative Information Agents. In *International Joint Conference on Artificial Intelligence (IJCAI)* (p. 37-42). Acapulco, Mexico.

Freitas, F. L. G., & Bittencourt, G. (2003). An ontology-based architecture for cooperative information agents. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence* (p. 37–42). San Francisco: Morgan Kaufmann Publishers Inc.

Gaizauskas, R., & Robertson, A. M. (1997). Coupling Information Retrieval and Information Extraction: A New Text Technology for Gathering Information from the Web. *IN proceedings of the 5th computed-assisted information searching on internet conference (RIAO'97)* (p. 356–370).

Girardi, C. (2007). HTMLCleaner: Exracting relevant text from web pages.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud., 43*(5-6), 907–928. doi:http://dx.doi.org.gate6.inist.fr/10.1006/ijhc.1995.1081

Huhns, M. (1994, Juin). *Distributed Artificial Intelligence for Information Systems*. Tutorial presented at Second International Conference on Cooperating Knowledge Based Systems (CKBS-94), Keele, UK.

Ireson, N., Ciravegna, F., Califf, M. E., Freitag, D., Kushmerick, N., & Lavelli, A. (2005). Evaluating Machine Learning for Information Extraction. Proceedings of the 22nd international conference on ML. doi:http://doi.acm.org/10.1145/1102351.1102395

Jade. (2006). *Java Agent DEvelopment Framework*. Accessed july 13th, 2010, from http://jade.tilab.com/

Jena. (2006). *Jena Semantic Web Framework*. Accessed july 13th, 2010, from http://jena.sourceforge.net/

Jess. (2006). *Jess the Rule Engine for the Java Platform*. Acessed July 13th, 2010, de http://www.jessrules.com/

Jung, J. J. (2007). Ontological framework based on contextual mediation for collaborative information retrieval. *Inf. Retr., 10*(1), 85–109. doi:http://dx.doi.org.gate6.inist.fr/10.1007/s10791-006-9013-5

Karkaletsis, V., Spyropoulos, C. D., Souflis, D., Grover, C., Hachey, B., Pazienza, M. T., et al. (2003). Demonstration of the CROSSMARC system. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations* - Volume 4. doi:http://dx.doi.org/10.3115/1073427.1073434

Kauchak, D., Smarr, J., & Elkan, C. (2004). Sources of Success for Boosted Wrapper Induction. *Journal of Machine Learning Research*, *5*, 499–527.

Kushmerick, N. (1999a). Gleaning the Web. *IEEE Intelligent Systems, 14*(2), 20–22. doi:http://dx.doi.org.gate6.inist.fr/10.1109/5254.757626

Kushmerick, N. (1999b). Gleaning the Web. *IEEE Intelligent Systems, 14*(2), 20-22. doi:http://dx.doi.org/10.1109/5254.757626

Lesser, V., Horling, B., Klassner, F., Raja, A., Wagner, T., & Zhang, S. X. (2000). Big: An agent for resource-bounded information gathering and decision making. *Artificial Intelligence*, *118*, 197–244. doi:10.1016/S0004-3702(00)00005-9

Lima, R., Espinasse, B., & Freitas, F. (2010). Adaptive Information Extraction System based on Wrapper Induction with POS Tagging. I*n Proceedings of SAC-ACM 2010, Sierre, Switzerland*.

Lorenzi, F., Santos, D. S., & Bazzan, A. L. C. (2005). Negotiation for task allocation among agents in case-base recommender systems: a swarm-intelligence approach. In E. Aimeur (Ed.), *Multi-Agent Information Retrieval and Recommender Systems - Nineteenth International Conference on Artificial Intelligence (IJCAI 2005)* (p. 23--27). Edimburgh, Scotland.

Maynard, D., Peters, W., & Li, Y. (2006). Metrics for evaluation of ontology-based information extraction. *WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON)*, Edinburgh, Scotland.

Mccallum, A., Nigam, K., Rennie, J., & Seymore, K. (1999). *Building Domain-Specific Search Engines with Machine Learning Techniques*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography*, *3*, 235–244. doi:10.1093/ijl/3.4.235

Muslea, I., Minton, S., & Knoblock, C. (1998). STALKER: Learning extraction rules for semi-structured Web-based information sources. *Proceedings of the AAAI-98 Workshop on "AI & Information Integration"*.

Newell, A., Shaw, J., & Simon, H. (1959). *Report on a General Problem-Solving Program*.

Nwana, H. S. (1996). Software Agents: An Overview.

Oates, T., Prasad, M. N., & Lesser, V. R. (1994). Cooperative information gathering: a distributed problem solving approach.

Pascal Challenge. (2005). Pascal Challenge. Accessed from http://nlp.shef.ac.uk/pascal/

Protégé. (2006). Protégé. *The Protégé Ontology Editor and Knowledge Acquisition System*. Accessed from http://protege.stanford.edu/

Siefkes, C., & Siniakov, P. (2005). An Overview and Classification of Adaptive Approaches to Information Extraction. *Journal on Data Semantics*, *IV*, 172–212.

Tang, J., Hong, M., Zhang, D., Liang, B., & Li, J. (2007). Information Extraction: Methodologies and Applications. Dans *Emerging Technologies of Text Mining: Techniques and Applications* (p. 1-33). Prado and Edilson Ferneda (Ed.), Idea Group Inc., Hershey, USA.

The Satine Project. (2006). The Satine Project. Accessed July 15th, 2010, from http://www.ve-forum.org/apps/pub.asp?Q=1275&T=Clusters%20and%20Projects

TIES. (2004). *TIES - Trainable Information Extraction System*. Accessed July 15th, 2010, from http://tcc.itc.it/research/textec/tools-resources/ties.html

Tufis, D., & Mason, O. (1998). Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC* (p. 589–596).

Turmo, J., Ageno, A., & Catala, N. (2006). Adaptive information extraction. *ACM Comput. Surv., 38*(2), 4. doi:http://doi.acm.org/10.1145/1132956.1132957

Woerndl, W., & Groh, G. (2005). A proposal for an agent-based architecture for context-aware personalization in the Semantic Web. In *Multi-Agent Information Retrieval and Recommender Systems Proceedings of, IJCAI-2005 Workshop, 2005.*