

Agent Systems' Influence on Information Retrieval

Abstract – This paper explains the possibilities and benefits of using modern heterogeneous agent systems to support users in finding appropriate information. Nowadays search engines try to expand by using some smart technologies and algorithms to support users information needs. But they are still far away from real intelligent assistance. The results delivered by the search engines are at best a starting point for further refinement. The given links have to be followed manually just to recognize that they are often useless. The sheer number of returned links makes it impossible for a human being to pay attention to every link. A case description shows how users can get help in solving this unsatisfying situation by delegating the search task to intelligent software agents.

Introduction

Wouldn't it be fascinating to have some intelligent software assistant working in the background on behalf of your interests? Supporting you in your everyday information needs? An assistant that has necessary a minimum of intervention by you?

The normal internet user is conducting his queries by using general search engines several times a day. The given result of this search engines has to be most probably refined again and again to get the desired information. Often this is a laborious task. Too much time must be spent to obtain the desired information. One of the big problems is the so-called information overload. Often the hits of the search engines go into the thousands. Agent concepts could fundamentally alter the nature of information systems. Intelligent software agents can take the burden of searching and filtering of appropriate information from the user. Thus the user has more time to concentrate on fulfilling his daily work.

In this work the author shows the benefits a user can gain using an agent system to support him in his information needs. The aim of the paper is also to give the reader an insight in what is possible and what can be expected from future developments in the area of information retrieval.

Scenario

With the following scenario the author wants to give an impression to what an intelligent agent system is capable or what a user can expect from a real intelligent agent system.

Let's have a look at user A. He wants to travel from Milano in Italy to Zurich in Switzerland. User A wants to participate in a conference that will be held at the university of Zurich. It's obvious that beside the search of a suitable hotel, which is

preferably near the conference place, he has also necessary to get some information about how to travel to Zurich. Some information agent is sent out to gather information about free hotels in Zurich and another agent collects information on how to travel to Zurich. User A is instructed about the distance and how fast he will be when travelling by train, car or plane. He even gets presented the costs of these different kind of travel. Since in Switzerland the currency is in Swiss francs, another agents gets active and calculates Swiss francs in euro, since user A is Italian and used to this currency - after all he would probably also like to know what amount the currency is in the old Lira. And what a surprise for user A as another agent which is informed by the search agent delivers some useful tips on how to spend the leisure time in Zurich. User A is informed about some musical which matches his personal preferences and is informed about a famous pizza restaurant since he likes best Italian food. And if this is not already enough user A gets the information about the address of his colleague in Zurich who will also participate at the conference. The intelligent agent fetches this information from user A's contact book. Last but not least user A is informed about the actual weather forecast. An abstract view of the scenario is shown in figure 1.

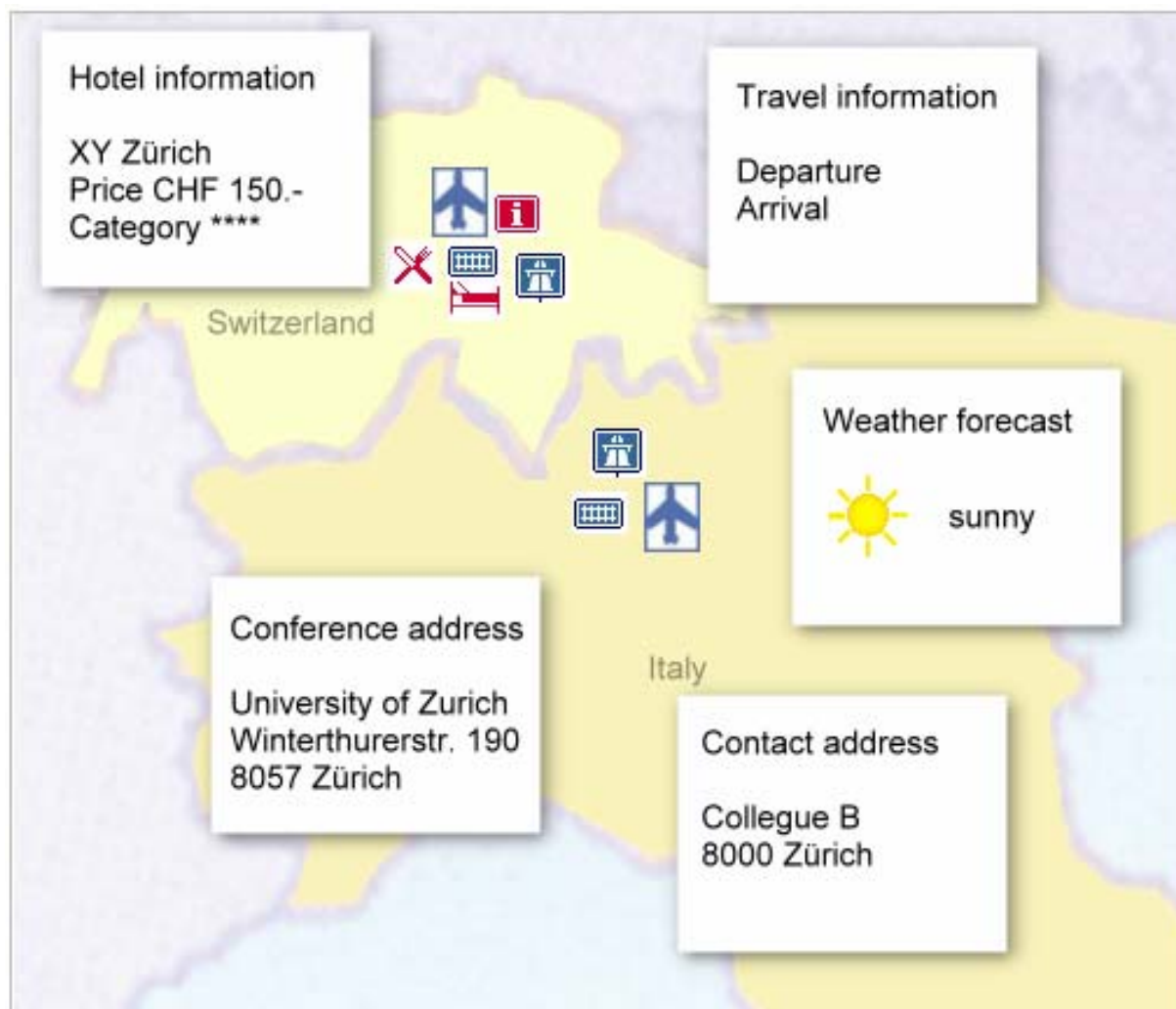


Figure 1 Scenario conference

Insights into the implementation

The previous scenario demonstrates the use of intelligent agents conducting a search on the internet and shows a wealth of benefits to a simple query using a normal search engine like Google. Today's applications do not assist the user very well in getting more out of a search query. They do not know what information a user is not interested in and filter it out. They do not know what data of the retrieved results the user most probably needs and they cannot group them correctly. Google has some simple mechanism correcting your query and trying to guess what you could have meant when it does not recognize a search term. Some engines show ads that are related to the query you made. But this are just some simple features. They can't predict what additional information needs a user has and trying to enrich the results with additional information. To some degree parts of the scenario are realized, mostly in experimental prototypes. There must be admitted that with today's web and the available information in HTML files, it is extremely difficult to automatically use the full potential. The next generation of the internet, the semantic web, promises to bring search tasks to their full potential. The following description gives an insight how such an agent systems can be implemented and what difficulties it has to conquer.

The system is coded in Java, since it is a wide used language, which offers many features when dealing with the internet. It is established both in universities and in industry. It has a great user community from which one can get fast and reliable support and there are many tools and components available, which make the development of applications easier.

What is needed for the system is an implementation built on an agent system where different agents can communicate, work together and move to other hosts to better and faster fulfil their tasks.

The topology of the internet is dynamic and the content is changing rapidly. Some of the techniques for dealing with the size and complexity of information systems are modularity, distribution, abstraction, and intelligence, e.g., being smarter about how you seek and modify information. (Huhns) That's why the use of a distributed agent system is best.

JADE is an excellent choice to be the underlying agent system, which holds all together. JADE is short for Java Agent DEvelopment Framework. It is a software framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middleware that claims to comply with the FIPA specifications. The agent platform can be distributed across machines and the configuration can be controlled via a remote GUI. (TILab)

Our agent system holds different types of agents that work together. Therefore it is a heterogeneous agent system. Each agent performs tasks on behalf of the user or other agents that act in the interest of the user.

The agent system contains the following kind of agents to support the user in his information needs:

- Interface agent
- Profiling agent

- Coordination agent
- Specialized agent for specific domain
- Specialized agent for information broker

In the following section the different kind of agents and their tasks are described.

The interface agent is like a personal assistant who is collaborating with the user. It supports and provides proactive assistance. The user is guided when setting up his query and asked for assistance if necessary. The aim of the interface agent is to provide the user with a clean view on the delivered results of the search and helps the user to further work on the results.

The interface agent works closely together with the profiling agent, which holds personal information about the user. The task of the profiling agent is to gather all relevant information that can help the user to be more satisfied with the search result. It observes and monitors the actions taken by the user in the interface. The collaborating agents give immediate feedback to the profiling agent. The agent learns through receiving positive and negative feedback from the user, by receiving explicit instructions from the user and by asking other agents for advice. Profiling agents can access collected information of other user profiles to categorize users need and to better assist users. (Nwana)

The coordination agent creates a search strategy and informs the search agents which tasks they have to fulfil. It coordinates the incoming results from the agents and sends them back to the interface agent.

The search agents are specialized agents that have knowledge about a specific domain. This knowledge is described in an ontology. An ontology helps agents to describe facts, beliefs, hypotheses and predications about a domain. Ontologies range in abstraction to very general terms to terms that are restricted to specific domain of knowledge. (Bergenti et al.)

Picking up the hotel part of the search scenario, the specialized agent knows how to get the internet addresses of sites holding information about a specific hotel in the desired destination area. It has knowledge about how to extract appropriate information from this sites and composes the information for the coordination agent. Another search agent is specialized on broker services. Broker services like expedia.com have a form with input fields that have first to be filled out before conducting a search on hotels. The great advantage of broker services is the fact, that they give back highly relevant results always in the same schema. Most broker services have access to a database with the desired information. Unfortunately only the broker service has access to this well structured data. Often in the results of the broker services there are links to the specific hotel site. If such a link is available the specialized agent for broker informs the coordination agent which orders a specialized agent of the hotel domain to visit this page.

The key problem is that the information is often only available in HTML format. This fact makes it very difficult for an agent to automatically extract the necessary information since the HTML format does not contain any semantic information. Semantically enriched pages that use RDF or XML are not many available on the web. So it is necessary for the agents do be able to deal with HTML pages since this

is the standard on the today's web. The agent has to conduct a semantically mapping to enable a comparison of the attribute values.

When looking at the search strategy of a human there can be noticed that available links on the page have to be considered and visited to gain good results. Agents do this task for the user and follow the available links on a page to check whether there are further information which could be of interest and report them back to the coordination agent, so duplicate visits of already scanned pages can be eliminated.

The coordination agent combines the result of the search agents and sends them back to the interface agent that presents the results to the user in the desired manner according to the user's profile. It will check for duplicated results, grouping similar entries and enable the user to refine his query. The users reaction on the results is closely observed and stored in the profiling agent. So future searches of the user can profit from the analysis of past searches.

User profiles and results of past searches are stored as XML files. XML files can easily be shared with other applications and do not have a great footprint like databases.

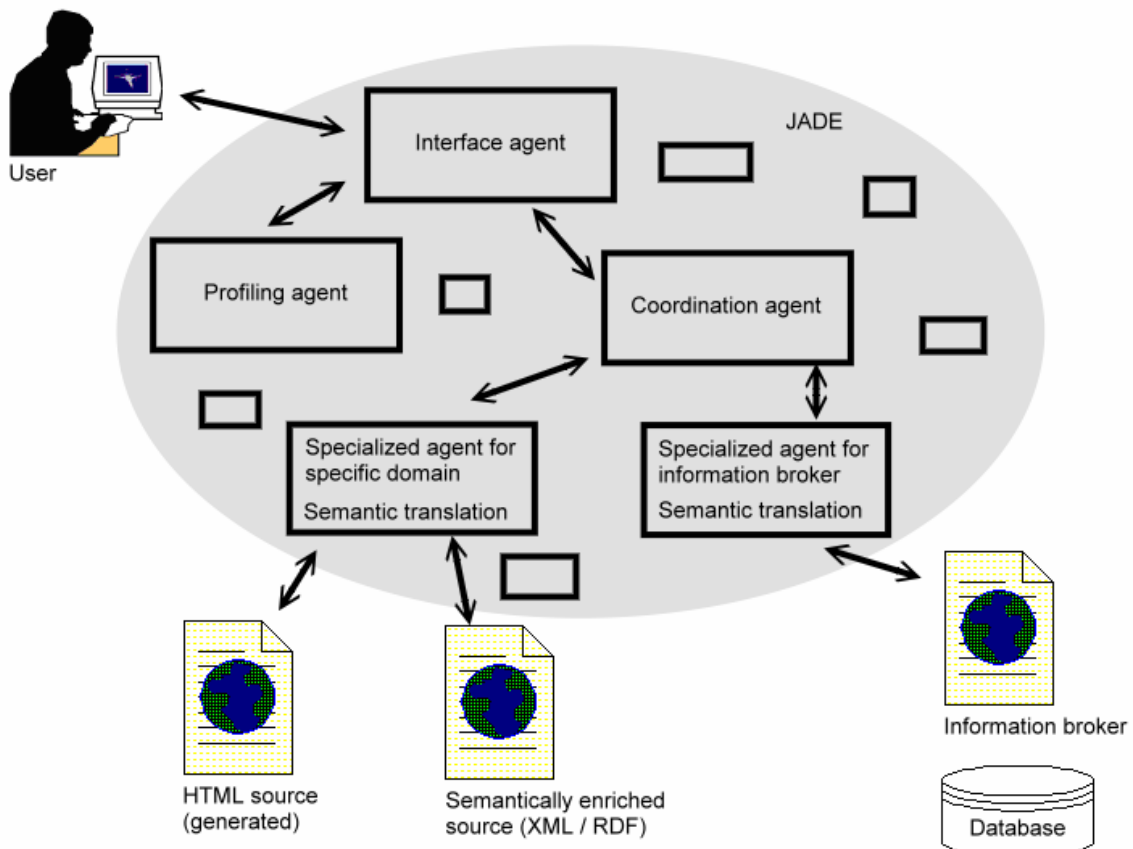


Figure 2 Logical view of the execution environment

Figure 2 shows the logical view of the execution environment. The JADE agent platform houses many more agents. Think of many other agent instances, which act together.

Problems

There are many more problems and difficulties that could be listed but the following list should give some impression with what kind of problems information agents have to struggle.

- Most of the available information on the web is not very well machine-readable. Since it is a goal of the work to assist the user in querying web pages that hold unstructured information the search agents have to deal with this problem. The agents have to make a semantic mapping so that the results can be more accurately filtered and understood by the user.
- Since a HTML document uses mainly only tags for presentation and no tags to express semantic meaning it is quite a heavy task to extract a simple address information out of “Foostreet 25”. In a XML document the address is embraced by some tags which help understand the meaning “<address>Foostreet 25</address>”. With this we gain some major benefit in presenting and understanding the information correctly.
- False HTML Code has also to be mastered. The quality of information differs. Some pages contain false information, are out of date or not available. There are also mechanisms to react on this problem.
- Often there is a semantic inconsistency between the many information different pages hold. Doing a manually search on four different broker services for hotels brought back more than 52 different attribute fields. One service gave quite detailed information another one only a few fields. The name of the same hotel can easily vary. For example service one writes “Swissotel”, service two “Swissotel Zurich Hotel International”, service three “Swissotel Zurich” and service four “Swissôtel Zürich”. The solution for this problem is to check the address. If it is the same address then it is the same hotel. But what when this information is not available? To illustrate some more difficulties in comparing hotels is to have a look at the hotel class. One service writes down “4 stars or more”, “5 stars” another one writes down “Standard” or “First Class”.
- Not all pages are accessible. Some require a valid login.

Advantages using an agent technology

By using an agent system based on JADE there must be some benefit about software based on non-agent systems. Otherwise there is no need changing existing systems to an agent environment. The following list of benefits is not finally.

Benefits for the user

- Increased productivity by automation. The agents perform repetitive tasks.
- The agent learns the users behaviour and gives proactive assistance. He adapts to the users information needs and habits.
- Off-line work possible, the agent can perform any task remotely. The user sends his query and logs off. If the agent is finished the user loges in and gets the result or he sees what time is still necessary to complete a task.

Benefits for the developer

- Gain true scalability
- Mobility supported by agent system
- Easy upgrading or updating of existing agents by just register the new agent to the system without needing to shut down the whole system
- Programming tasks can more consequently be shared among developers. By using the specifications a new agent can be implemented or the capabilities of an existing agent can be powered up.

Discussion

The conference scenario is one of many possible scenarios regarding intelligent search systems. Using agents can be a great benefit to the user. Agent systems will be a key feature of future information systems.

Another aspect of the author's work is to gain insight in the benefit of the so-called Semantic Web that enables computers to better understand the available information. In the above implementation chapter there was mentioned that there lies a great potential in the future semantic web to be a truly gain in extending web searches. If an agent gets access to a semantic enriched file, e.g. a XML file, with specific semantic tags the extraction of accurate information is extremely simplified for an agent and can be automated to a high degree. To what extend semantic information is applicable in the near future to software agent or other information retrieval applications is unsure. There is a too big amount of documents available to be changed to a semantically enriched document in a timely manner. Probably the semantic information in the XML document is not available to the public and therefore cannot be accessed through the web browser. Instead a HTML document is generated by using a XSL file. And then we have again the same situation that we have now: the software agent needs to extract information from a HTML document.

One important question for the research in this area is to what extend an agent can explore an unknown domain and extract desired domain attributes. What elementary information is needed for the agent to do the job?

Another theme that has to be cared for and is not explicitly mentioned in this paper is security. For example there has to be considered who has access to the user

profiles, what information is stored in the profiles and how is a profile protected against unauthorized access.

The implementation with JADE is still in development but it's promising to deliver some interesting results. Deriving parallels to nature by regarding a simple ant that cannot fulfil great tasks a whole colony can reach impressive things like building an ant-hill. The whole colony is more than the sum of it's ants. So will be the agent system: the whole interaction of all agents is more than the sum of all single agents.

Glossary

Browser

A web client that enables a human to read information on the internet.

HTML

The language that most web pages are currently written in.

JADE Java Agent DEvelopment Framework

Software framework for the implementation of multi-agent systems.

RDF Resource Description Framework

A framework for constructing logical languages. RDF provides a mechanism to exchange metadata.

Semantic Web

The Web of data with meaning in the sense that a computer program can learn enough about what the data means to process it.

XML eXtensible Markup Language

Generic language for creating new markup languages. XML documents consist mainly of text and tags, and the tags imply a tree structure upon the document.

XSL eXtensible Stylesheet Language

A style language for XML also allowing document transformation.

References

Bergenti, F., Caire, G., Pels, R., van Aart, C. (2002): Creating and Using Ontologies in Agent Communication. In: *exp, in search of innovation*, Vol 2 (3), 110-121.

Huhns, M. N., Singh, M. P. (1998): Managing Heterogeneous Transaction Workflows with Co-operating Agents. In: R. J. Nicholas, M. J. Woolridge (Eds.) *Agent Technology. Foundations, Applications, and Markets*. Springer, Berlin Heidelberg, 221-239.

Nwana, H. S., Ndumu, D. T. (1998): A Brief Introduction to Software Agent Technology. In: R. J. Nicholas, M. J. Woolridge (Eds.) Agent Technology. Foundations, Applications, and Markets. Springer, Berlin Heidelberg, 29-45.

TILab S.p.A. (formerly CSELT): JADE. Torino, Italy. 5 June 2003
<<http://sharon.cselit.it/projects/jade>>

Author

Contact

First name: Christoph

Last name: Kindle

Address business:

ICS Management GmbH

Maschlinastr. 25

9495 Triesen

Principality of Liechtenstein

Phone business: +423 / 399 32 50

E-Mail: ckindle@ics.li

Group homepage of university: <http://www.ifi.unizh.ch/egov/>



Curriculum vitae

Christoph Kindle received his Master's Degree in Computer Science and Business Administration from Zurich University, Switzerland in 1998. During his studies he made a practical course as project leader at the Union Bank of Switzerland. Since November 1996 he is working at ICS Management GmbH in the Principality of Liechtenstein where he could later engage as partner. Since March 1999 he is working as an external PhD-Student on a thesis in the area of intelligent software agents at the Department of Information Technology of the University of Zurich.